

**TEMPERATURE DETECTION FOR FEVER PATIENTS
USING RFID AND GSM MODEM**

by

SHARIFFUDIN BIN SAHEDAN

FINAL REPORT

**Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)**

**Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan**

© Copyright 2009

by

Shariffudin bin Sahedan, 2009

CERTIFICATION OF APPROVAL


TEMPERATURE DETECTION FOR FEVER PATIENTS USING RFID AND GSM MODEM

by

Shariffudin bin Sahedan

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:



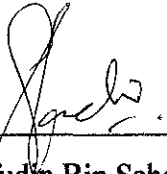
(Pn Hanita Daud)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

December 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



Shariffudin Bin Sahedan

ABSTRACT

Life-threatening fever such as Malaria and Typhoid are treated manually at most of the hospitals nowadays. The nurses have to check the temperature and pulses of suspected victims at certain hours in a day. What makes it dangerous is that in between the interval hour, the patient is all alone without supervision from the nurse or the doctor. If at this time the temperature of the patient is suddenly rising over 39°C, it can be fatal. Besides, patient's blood sample needs to be taken when his/her body temperature is at the highest level to confirm the diagnosis of the disease and identify the infectious parasites. This project "Body temperature detection system for malaria and typhoid fever patient using RFID and GSM network" comes into picture. As the temperature of the patient exceeds the limits, RFID active tag will be triggered, communicate with the RFID reader at nurse's station, notify nurse through alert popup and then, nurse will SMS to the doctor on duty using GSM Modem. With this, the patient will receive the right treatment at the right time.

ACKNOWLEDGEMENTS

Firstly I would like to praise Allah the Almighty, which have helped and guided me in completing my Final Year Project (FYP) in Universiti Teknologi PETRONAS. I also like to convey my deepest appreciation to my family, for their support, giving me motivation and advices, understanding, and their encouragement for me to be a better human.

My almost gratitude goes to my project supervisor, Ms. Hanita Daud for her supervision and support throughout the process of the project. I would also like to express my full appreciation towards Director of Hospital Sg. Buloh, Dr. Khalid Ibrahim for his support and advice for the success of this project.

In additional, my thanks also go to my fellow friends and individuals who were supportive throughout the project period. The support and encouragement from the people above will always be pleasant memory throughout my life.

TABLE OF CONTENT

ABSTRACT	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENT	vi
LIST OF FIGURES	ix
LIST OF TABLES	x
CHAPTER 1 INTRODUCTION	1
1.1 Background Study	1
1.2 Problem Statement	2
1.3 Objectives.....	2
1.4 Scope of Study	3
<i>1.4.1 Analysis the symptom of Malaria and Typhoid fever</i>	<i>3</i>
<i>1.4.2 Perform analysis of RFID tag and reader suitable for the project</i>	<i>3</i>
<i>1.4.3 Built a suitable GUI with Visual Basic (VB) and database system with MySQL</i>	<i>3</i>
<i>1.4.4 Test at the hospital.....</i>	<i>3</i>
1.5 Output of the system	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Malaria	5
2.2 Typhoid	6
2.3 Overview of RFID.....	7

2.4	RFID-Tags	8
2.5	Reader/Antenna.....	10
2.6	Operating Principles of RFID	12
2.6.1	<i>Inductive Coupling</i>	12
2.6.2	<i>Backscatter Coupling</i>	13
2.7	Operating Frequencies	14
2.8	Overview of GSM Modem	16
2.9	RFID Applications	17
2.9.1	<i>RFID temperature/ID tags could help spot bird flu [18]</i>	17
2.9.2	<i>Application in monitoring and traceability of hemoderivates [19]</i>	18
2.9.3	<i>Trailer temperature alert [20]</i>	18
2.9.4	<i>RFID Pill Monitors Body Temperature [21]</i>	19
CHAPTER 3 METHODOLOGY		20
3.1	Procedure Identification	20
3.1.1	<i>Phase one (Project Initialization)</i>	21
3.1.2	<i>Phase two (Designing)</i>	21
3.1.3	<i>Phase Three (Prototyping)</i>	21
3.2	Project Overview	22
3.3	RFID Hardware and Specification	23
CHAPTER 4 RESULTS AND DISCUSSION		26
4.1	Interface	26
4.1.1	<i>Interface for System Login and Main Menu</i>	26
4.1.2	<i>Interface for Administration</i>	27
4.1.3	<i>Interface for Patient Record</i>	29

4.1.4	<i>Interface between Database and GSM Modem</i>	30
4.2	Demo in front of Sg. Buloh Hospital's Director.	31
4.3	Testing the Device	32
4.3.1	<i>Range between each tag and reader</i>	32
4.3.2	<i>User friendly interface</i>	32
4.3.3	<i>Error percentage of the system</i>	32
4.4	Lab Test	33
4.5	Hospital Tests	34
4.5.1	<i>Patient 1</i>	34
4.5.2	<i>Patient 2</i>	35
4.6	Test Result	36
4.7	Challenges faced	37
4.7.1	<i>Building a source code for the system</i>	37
4.7.2	<i>Building the database for the system</i>	38
4.7.3	<i>Integration between RFID hardware and GSM modem</i>	38
4.7.4	<i>Hardware Reliability</i>	39
CHAPTER 5 CONCLUSION AND RECOMMENDATIONS		40
5.1	Conclusion	40
5.2	Recommendations	40
REFERENCES		41
APPENDICES		43
APPENDIX A Database Table		44
APPENDIX B Visual Basic Source Code		48

LIST OF FIGURES

Figure 1: Flow of the System.....	4
Figure 2: Basic RFID Tag Circuit Diagram.....	8
Figure 3: Basic RFID Reader Diagram.....	10
Figure 4: Basic Configuration of Reader and Tag Antenna in RFID Application.....	11
Figure 5: RFID Reader.....	11
Figure 6: GSM Modem	16
Figure 7 : Flow Chart.....	20
Figure 8: Project Overview	22
Figure 9: System flow for ActiveWave hardware used	24
Figure 10 : Active Wave RFID Kit.....	25
Figure 11: Login Menu	26
Figure 12: Main Menu	27
Figure 13: Administration Menu.....	27
Figure 14: New User Registration Menu	28
Figure 15: New Patient Registration Menu.....	28
Figure 16: Database for New Patient Registration section	29
Figure 17: Patients Record.....	29
Figure 19: Alert Form	30
Figure 18: Example of Patient Record	30
Figure 20: SMS Alert Form	31
Figure 21: Author with Project Supervisor at Hospital Sg. Buloh.....	31
Figure 22: Temperature vs. time for lab testing.....	33
Figure 23: Patient's 1 Temperature vs. Time.....	34
Figure 24: Patient's 2 Temperature vs. Time.....	35
Figure 25: Flow diagram of the system.....	37

LIST OF TABLES

Table 1 : Table of Malaria cases reported in Malaysia	6
Table 2 : Table of Typhoid cases reported in Malaysia	7
Table 3: Eight Bands in Radio Frequency Spectrum	14
Table 4: RFID Frequencies, Uses and Typical Range	15
Table 5: Setting ActiveWave Reader field strength ratio corresponds to estimated distance	24

LIST OF ABBREVIATIONS

RFID	RADIO FREQUENCY IDENTIFICATONS
GSM	GLOBAL SYSTEM FOR MOBILE
RF	RADIO FREQUENCY
VB	VISUAL BASIC
GUI	GRAPHIC USER INTERFACE
SMS	SHORT MESSAGING SERVICE
SIM	SUBSCRIBER IDENTITY MODULE
AT	ATTENTION COMMANDS
PCMCIA	PERSONAL COMPUTER MEMORY CARD INTERNATIONAL ASSOCIATION
USB	UNIVERSAL SERIAL BUS

CHAPTER 1

INTRODUCTION

1.1 Background Study

This project is basically about a system that can notify nurses immediately after temperature rise from remote location involving Malaria and Typhoid fever patients in the hospital and also alerting the ideal time to take blood sample for diagnosis. These patients are known to have irregularities in body temperature. The ideal time to take blood sample from them is when the temperature is at its highest level. By using the system, whenever the body temperature is exceeding the limit that has been set, the system will be activated. The system uses Radio Frequency Identification (RFID) and GSM network for the project to achieve its objectives.

RFID is a wireless technology that has an ability to identify, locate, track, and monitor people and object between the tag antennas and the reader. RFID reader obtains the information of objects and surrounding through communication with RFID tag antennas. These antenna and transceiver read the radio frequency and transfer the information to a processing device, and a transponder, or tag, which is an integrated circuit containing the RF circuitry and information to be transmitted.[1]

So, for this project, the RFID tag will include temperature sensor on it and having a dedicated threshold value. Whenever temperature is exceeding the threshold value, it will be triggered and responding to the RFID reader. Then, RFID will send signal of the triggered location to the PC and alert the nurse. From here, GSM modem will take charge where nurse will send notifications via SMS to doctor's on duty.

1.2 Problem Statement

Malaria and typhoid fever patient suffers from fever spikes; where the body temperature rise and fall in a sudden. Patients of these two fevers need extra supportive care and the only physical sign that doctor can have from a suspected patient is fever. The fever is initially persistent. The doctor or nurse usually has to check the patient every four hours to take his/her temperature. When the temperature of the fever is around its peak (39-40°C), that is the time to take blood sample to check whether there are active parasite or bacterium that are related to malaria or typhoid. At that temperature, the patient is potentially become unconscious and the body may become over- hydration or under-hydration. So, nursing care is important at that time. The late and lack of treatment at that period of time may become life-threatening to the patient.

The main concern here is that the doctor cannot predict when the temperature of the patient exceeds its limit and the patient are all alone when his or her temperature is over the limit. So, by creating this new system, it will solve this problem where it can assure that the patient will receive the quickest treatment as alert will popup at the nurse's station and notifications via SMS will be send to the doctors on duty whenever the patient's body rises over its limit. Besides, the system will ease the nurse from attending to the patients to take their body's temperature because the ability of the system that may be adjusted to measure temperature at specified interval as required.

1.3 Objectives

The main objective of this project is to focus on application of RFID and GSM in medicine. The other objectives are to design a system that can;

- i- automatically measure body temperature for certain preset threshold value and record them into patient's database for monitoring purposes.
- ii- alert the nurse at her station and send SMS to the doctor whenever patient's core body temperature reach the peak of the fever (39°C).

1.4 Scope of Study

1.4.1 Analysis the symptom of Malaria and Typhoid fever

In developing this system, the author needs to understand theoretically and also the manual process involve at the hospital for these two types of fever. From here, the author will know about things and information needed to develop the system.

1.4.2 Perform analysis of RFID tag and reader suitable for the project

In designing the system, the author should know about the current manual set up work used in hospital nowadays. In designing this system, some criteria should be considered. The first one is the range of communication between the tag at the patient and the receiver at the nurse's station. The second one is the ability of the system to operate between the ranges of temperatures and detect the rise of the temperature from normal condition.

1.4.3 Built a suitable GUI with Visual Basic (VB) and database system with MySQL

A suitable database is needed besides the communication technologies used. Database with doctor's and patient's information must be clear and easy to retrieve by VB. Thus, GUI of the system need to be user friendly so that the nurse can easily adapt with this new system.

1.3.4 Test at the hospital

After developing the prototype, this system will be tested at hospital. From there, further integration and modification will be made to make the system runs smoothly.

1.5 Output of the system

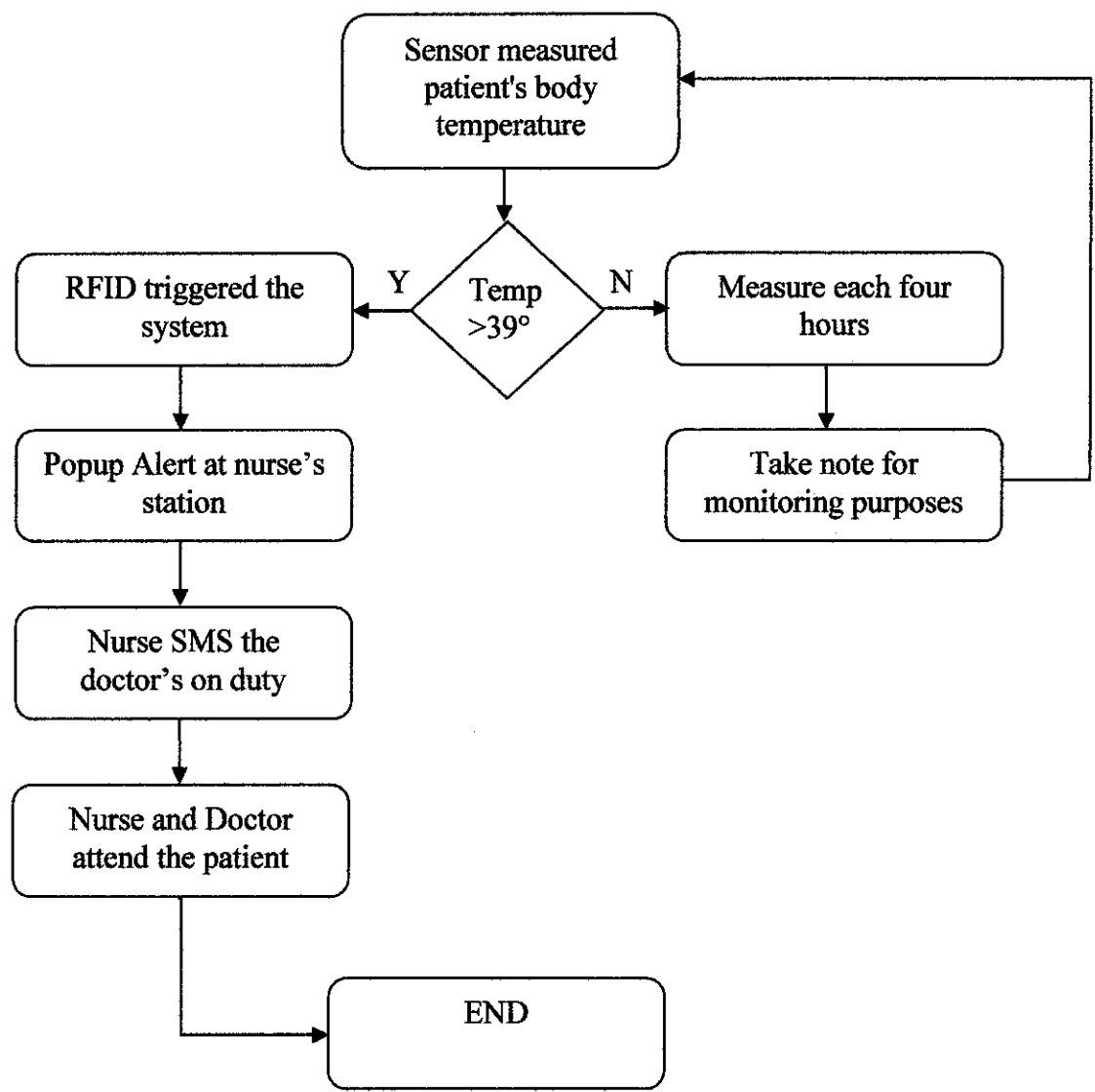


Figure 1: Flow of the System

CHAPTER 2

LITERATURE REVIEW

2.1 Malaria

What is malaria? Malaria is one of the planet's deadliest diseases and one of the leading causes of sickness and death in the developing world. It is a disease caused by a parasite called as *Plasmodium*, which infects red blood cells. There are four species of malaria; the serious type is *Plasmodium falciparum* malaria. It can be life-threatening. The other three species of malaria (*P. vivax*, *P. malariae*, and *P. ovale*) are generally less serious and are not life-threatening. [2]

How we can be infected by these parasites? The infection involves two hosts; humans and mosquitoes. The disease is transmitted to humans when an infected mosquito bites a person and injects the malaria parasites into the blood. These parasites travel through the bloodstream to the liver, multiply there and then infect the human red blood cells. [3]

Symptoms of malaria include fever as high as 39° to 40° C, headache, pains elsewhere in the body and also vomiting; usually appear between 10 and 15 days after the mosquito bite. On physical examination, fever is the only sign to detect malaria patient. If the doctor missed the physical sign and the patient is not treated, malaria can quickly become life-threatening by disrupting the blood supply to vital organs.[4]

According to the Ministry of Health Malaysia [5], the incidence rate (per 100 000 populations) of malaria as below:

Table 1 : Table of Malaria cases reported in Malaysia

Year	No of case
1999	25.90
2000	57.20
2001	53.22
2002	44.91
2003	25.28
2004	24.56
2005	21.31
2006	19.87

2.2 Typhoid

The second disease that we will discuss and research in the project is typhoid or also known as enteric fever. It is a serious infection that is caused by the bacterium *Salmonella typhi* (S. Typhi). [6] This bacterium only lives in humans.

How can a person get this disease or in other word, how the disease is transmitted? Anyone can get typhoid fever if they eat food or drink beverages that have been handled by a person who is shedding *S. Typhi* or if sewage contaminated with *S. Typhi* bacteria gets into the water used for drinking or washing food. [7]

Once *S. Typhi* bacteria are eaten or drunk, they multiply and spread into the bloodstream. The body reacts with sustain fever as high as 39° to 40° C and other signs and symptoms. The other sign and symptoms are; constipation or diarrhea, loss of appetite, rose-colored spots on the trunk, and an enlarged spleen and liver. Relapses are common. Symptoms are generally appearing one to three weeks after exposure.

According to the Ministry of Health Malaysia[5], the incidence rate (per 100 000 populations) of Typhoid as below:

Table 2 : Table of Typhoid cases reported in Malaysia

Year	No of case
1999	3.70
2000	3.40
2001	2.89
2002	3.48
2003	3.13
2004	1.93
2005	4.10
2006	0.77

2.3 Overview of RFID

For this project, RFID is one of the main technology that being used. RFID is the member of Automatic Identification and Data Capture (AIDC) technologies. It is fast and reliable means of identifying just about any material object. There are three basic components in RFID system, which are tags, readers, and antenna.

The tag contains at least two parts; Integrated Circuit (IC) and antenna. IC is for storing and processing information, modulating and demodulating a radio frequency (RF) signal while antenna is for receiving and transmitting the signal. There are two types of RFID systems, passive and active.

The reader is a device that provides network connectivity between tag data and interface. The reader is also known as an interrogator and it communicates with the RFID tags within its field of operation, performing any number of operations including simple continuous inventorying, filtering (searching for tags that meet certain criteria), writing to selected tags, etc.

2.4 RFID-Tags

RFID tags can be classified into three different types; active, passive and semi passive. All of these types have their own operating characteristics and the means by which they receive power for transmission determines their type.[8] Figure below shows the basic RFID Tag Circuit Diagram. RFID tags.

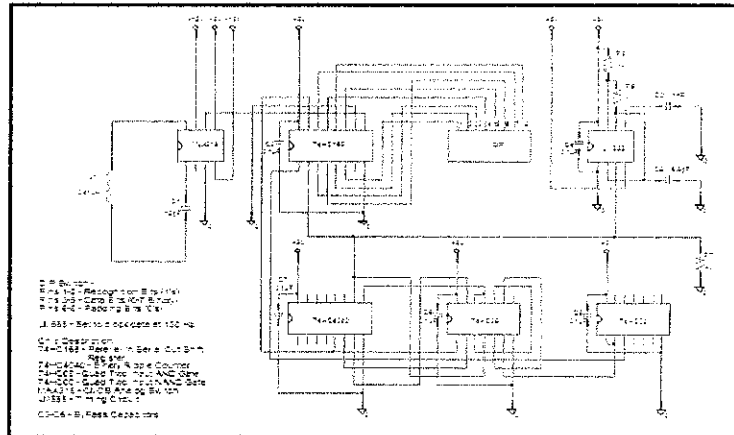


Figure 2: Basic RFID Tag Circuit Diagram

Active RFID Tags are powered by an internal battery and are typically read/write, where tag data can be rewritten and/or modified. It has a long range communication approach that has a reading distance between 50 to 100 m (150 to 300 feet). The battery supplied power of an active tag generally gives it a longer read range. The trade off is greater size, greater cost, and a limited operational life.[9]

Passive Tags operate without a separate external power source and obtain operating power generated from the reader. The advantages of passive tags are lighter than active tags, less expensive, and offer a virtually unlimited operational lifetime. The disadvantages are that they have shorter read ranges, typically a few feet at most than active tags and require a higher-powered reader. Read-only tags are typically passive and are programmed with a unique set of data (usually 32 to 128 bits) that cannot be modified.

Semi passive tags use battery to run the chip's circuitry, but communicate by drawing power from reader [10]. Once triggered by the reader's RF signal, these tags employ their own powers drained from the battery to perform their task.

Basically there are five differences in characteristics between active and passive RFID tags which are [11];

i. Differences in the Technical Characteristics

Active tag features two additional components compared to a passive tag that are, an on board power supply and on board electronics. The on board electronics consisting of sensors, microprocessors and I/O units are powered by the on board power which enables it to transmit the data to the reader on its own. With a continuous availability of tag power, the required signal strength from the reader to the tag is lower than what is required for a passive RFID tag. Also, the available signal strength is higher from a tag to a reader in an active tag.

On the other hand, Passive tags contain an integrated chip or circuit to absorb radio frequency waves from reader's signals and to send and receive data, low frequency or high frequency antenna and a plastic or mylar substrate which holds the pieces of the tag together. Since there is no battery, the power is supplied by the reader which draws the radio waves through the antenna forming a magnetic field. This allows the supply of the power to the tag which is restricted to the field of the reader.

ii. Communication Range

Passive RFID tags are constrained by the need for strong signals to power the tag and the small amount of power to respond to the reader, the communication range of a passive tag is limited to 3 meters or less. Active tags do not have the constraint of power and can thus transmit to as far as 100 meters or more.

iii. Data Storage

Active RFID tags have large read and write data storage almost 128 kilobytes and sophisticated data search and access capabilities while passive RFID, the data storage is less than 128 bytes with no search capabilities or data manipulation features.

iv. *Sensor Capabilities*

Active RFID tags are able to continuously monitor and record sensor input, while passive tags are only able to monitor and record sensor input when the tag is powered by the reader.

v. *Multi Tag Collection*

Passive RFID tag can collect 100 or so tags, at the range from only a single reader while an active tag can collect more than 100 tags from a seven acre region using a single reader.

2.5 Reader/Antenna

RFID reader is used to interrogate with RFID tags. It contains a module (transmitter and receiver), a control unit and a coupling element (antenna). The functions of the reader are: energizing, demodulating and decoding. Energizing is a process when an RFID reader emits a low-power radio wave field which is used to power up the tags. Demodulation is when the reader extracts the original information-bearing signal from modulated carrier wave by RFID tags. Decoding is the process where turning the raw incoming data received from RFID tag into meaningful context information for further processing and subsequent actions. It can be fitted with an additional interface that converts the radio waves returned from RFID tag into a form that can be passed on to another system such as computer or any programmable logic controller [12]. Figure below shows the basic RFID reader diagram.

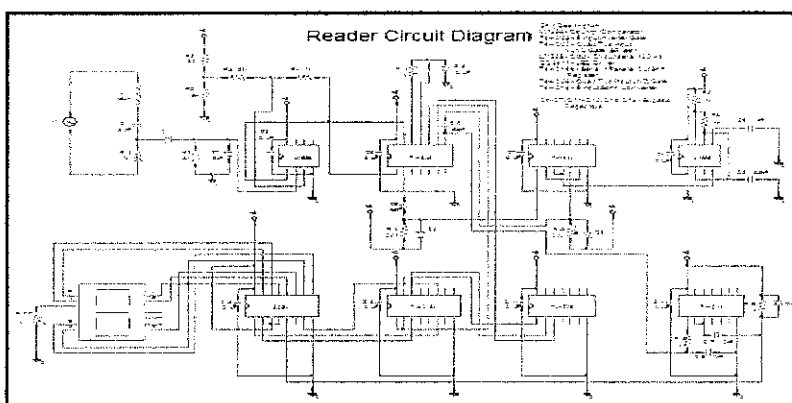


Figure 3: Basic RFID Reader Diagram

RFID reader has anti-collision algorithms that permit simultaneous reading of large number of tagged objects while ensuring that each tag is read only once. RFID reader obtains the information of objects and surroundings through communication with RFID tag antennas.

The reader uses its antenna to send digital information encoded in a modulated waveform as well as the tone that the tag uses to power itself. A receiver circuit on the tag is able to detect the modulated field, decode the information, and use its own antenna to send (backscatter) a response. Readers may be fixed (dock door or shelf installations), mobile (installed on a forklift or hand-held), or in the form of a module contained within a printer/encoder.

An antenna converts electrical current into electromagnetic waves that are then radiated into space in a particular pattern at a given level of intensity. A linearly polarized antenna radiates entirely in one plane in the direction of signal propagation, while with a circularly polarized antenna, the plane of polarization rotates in a circular fashion (effectively a corkscrew when considered in time), making a complete revolution during one period of the wave. Below figures shows the basic configuration of reader and tag antenna in any RFID application [13] and RFID reader itself.

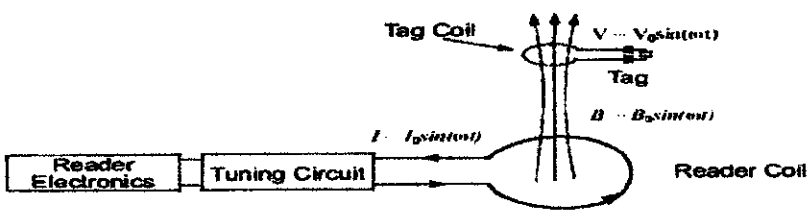


Figure 4: Basic Configuration of Reader and Tag Antenna in RFID Application.

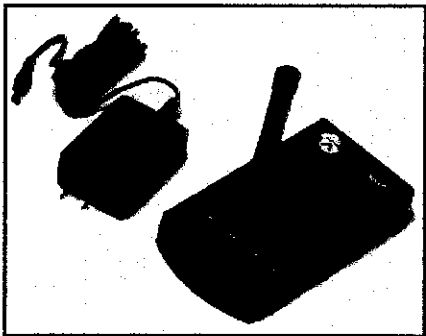


Figure 5: RFID Reader

2.6 Operating Principles of RFID

2.6.1 Inductive Coupling

Inductive coupling means that the tag and the reader's antenna are coupled by the magnetic flux through both coils. Inductively coupled transponders are almost always operated passively where all the energy needed for the operation of the microchip has to be provided by the reader. For this purpose, the reader's antenna coil generates a strong, high frequency electro-magnetic field, which penetrates the cross -section of the coil area and the area around the coil. Electro-magnetic field may be treated as a simple magnetic alternating field with regard to the distance between transponder and antenna due to the wavelength of the frequency range used ($< 135 \text{ kHz}$: 2400 m, 13.56 MHz: 22.1 m) is several times greater than the distance between the reader's antenna and the transponder.

This inductively coupled system is based upon a *transformer-type coupling* between the primary coil in the reader and the secondary coil in the transponder. This is true when the distance between the coils does not exceed 0.161, so that the transponder is located in the *near field* of the transmitter antenna.[14]

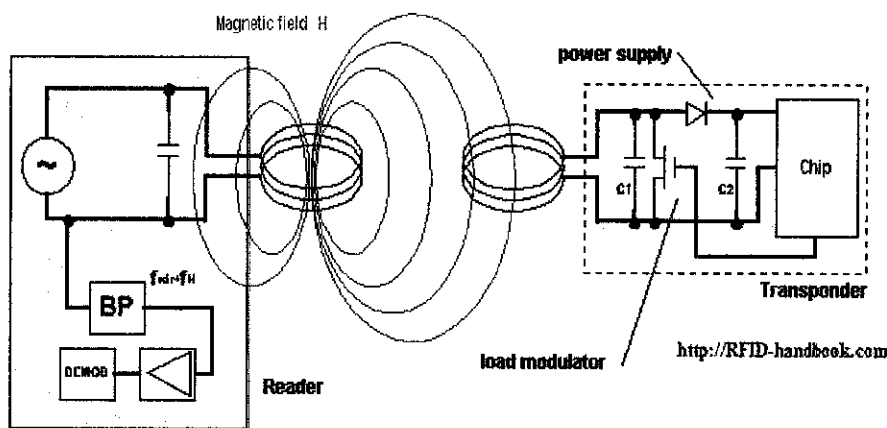


Figure 3: Operation Principle of Inductive Coupling

2.6.2 Backscatter Coupling

The electromagnetic waves from reader are reflected by objects with dimensions greater than around half the wavelength of the wave; in this case transponder. The efficiency with which an object reflects electromagnetic waves is described by its reflection cross-section. Objects that are in resonance with the wave front that hits them, as is the case for antenna at the appropriate frequency for example, have a particularly large reflection cross-section.

As in Figure 5 below, power P_1 is emitted from the reader's antenna, a small proportion of which (free space attenuation) reaches the transponder's antenna. The reflected signal travels into the antenna connection of the reader in the backwards direction and can be decoupled using directional coupler.

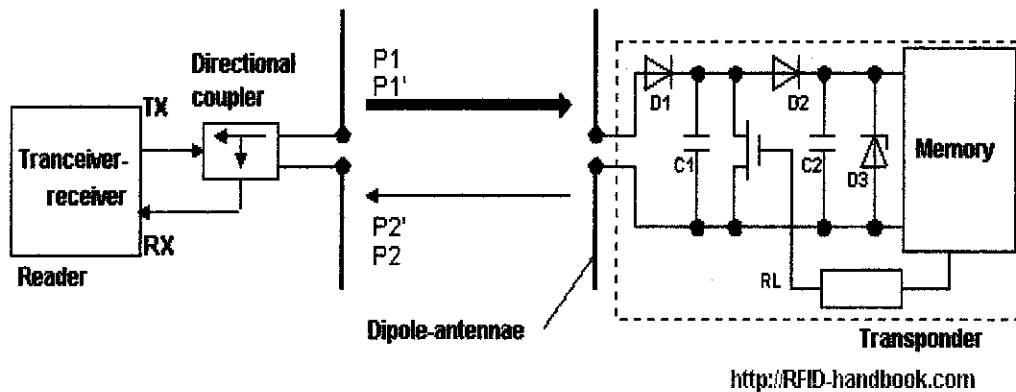


Figure 5: Operation Principle of a Backscatter Transponder

2.7 Operating Frequencies

RFID operates in unlicensed spectrum space, where its operating frequencies are refers to the size of the radio waves used to communicate between the RFID systems components. It is generally safe to assume that a higher frequency equates to a faster data transfer rate and longer read ranges, but also more sensitivity to environmental factors such as liquid and metal that can interfere with radio waves.

The operating frequencies are divided into several ranges or bands. With the exception of the lowest-frequency segment, each band represents an increase of frequency corresponding to an order of magnitude (power of 10). Table 3 depicts the eight bands in radio frequency spectrum, showing frequency and bandwidth ranges [15].

Table 3: Eight Bands in Radio Frequency Spectrum

Designation	Abbreviation	Frequencies	Free-space wavelength
Very Low Frequency	VLF	9kHz-30kHz	10km-100km
Low Frequency	LF	30kHz-300kHz	1km-10km
Medium Frequency	MF	300kHz-3MHz	100m-1km
High Frequency	HF	3MHz-30MHz	10m-100m
Very High Frequency	VHF	30MHz-300MHz	1m-10m
Ultra High Frequency	UHF	300MHz-3GHz	10cm-100cm
Super High Frequency	SHF	3GHz-30GHz	1cm-10cm
Extremely High Frequency	EHF	30GHz-300GHz	1mm-10mm

RFID systems currently operate in the Low Frequency (LF), High Frequency (HF) and Ultrahigh Frequency (UHF) bands. Each frequency has advantages and disadvantages relative to its capabilities. Table 4 shows a quick cross-section of the types of RFID technologies out there, their uses and their typical read ranges [16].

Table 4: RFID Frequencies, Uses and Typical Range

Frequency	Use	Pros and Cons	Range
125 kHz- 148 kHz Low Frequency (LF)			
Type: Passive	Animal Tracking, access control and Original Equipment Manufacturer (OEM) Applications	Signal negotiates liquids and metals fairly well. Higher tag cost due to long length solid copper antennas	½ to 4” is typical. 6” to 12” or maybe possible with specialized equipment.
13.56 MHz High Frequency (HF)			
Type: Passive	Anti-theft, document management, access control and OEM Applications	Antennas can be printed on substrate/labels, lowering tag costs. Serious interference from metals.	Can range from inches to several feet depending on reader hardware and tag type.
433 MHZ (and 2.5 Ghz) Ultra High Frequency (UHF)			
Type: Active	Highway toll payment systems, vehicle management, asset tracking, etc.	Very long range. Very high tag cost. Uses battery, so tags have finite lifespan (typically 5 years)	Typically around 30 feet, but can range up to hundreds of feet.

2.8 Overview of GSM Modem

Global System for Mobile Communication (GSM) modem is a wireless modem that works with GSM network. Wireless modem behaves like a dial-up modem but the main difference between them is that dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves. The number of SMS messages that can be processed by a GSM modem per minute is very low, which is only about six to ten SMS messages per minute[17].

A GSM modem can be an external device or a PC Card / PCMCIA Card. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate. Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the AT commands, this modem can have a function as below;

- i. Read, write and delete SMS message
- ii. Send SMS messages
- iii. Monitor the signal strength
- iv. Monitor the charging status and level of battery
- v. Read, write and search phone book entries

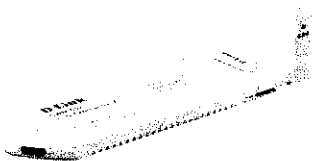


Figure 6: GSM Modem

2.9 RFID Applications

2.9.1 RFID temperature/ID tags could help spot bird flu [18]

Company named, Digital Angel Corporation had introduce its patented 'Bio-Thermo' temperature-sensing, implantable RFID chip and identity system to international poultry/bird markets with an initial geographic emphasis in Asia - with a view to helping poultry breeders spot early signs of disease such as Avian Flu. This company for now has focused its implantable RFID temperature sensing system on the pets and horses.

The identity system includes the Bio-Thermo chips and related ID and temperature-sensing RFID scanners that aim to provide early identification of temperature increases in individual birds. With this, it may in turn help with the identification and control of outbreaks of avian diseases.

With early detection of temperature increases, identifying and controlling disease outbreaks become important; such monitoring could help farmers avoid the complete destruction of their stock by identifying, quarantining and monitoring suspected cases at an individual level.

Besides using it for the birds, this Bio-Thermo is also being used in pets. The chip will be put into the pet's body and be registered specifically to the owner. The benefit is that it can take accurate reading of the pet's core body temperature.

2.9.2 Application in monitoring and traceability of hemoderivates [19]

There is a system used by hospitals to secure and monitor the temperature of their blood bank. Whenever the temperature of the blood bank is over its limit, the system will send alarm for the hospital to take action. It is also used in blood bag that they wanted to be transfer to somewhere else. This system provides product quality assurance before the blood bags are used by discarding the ones that turn out to be defective because of insufficient conservation standards. So, the design of this new system is for hemoderivate product storage and production technology. In effect, this will avoid many blood transfusion risks.

2.9.3 Trailer temperature alert [20]

This temperature alert is mainly used in chilled or refrigerated trailer. It has an in-can cab radio receiver incorporating a clear four digit LED display and temperature alert, linked to a temperature transmitter fitted within the trailer. The receiver is preprogrammed with 'acceptable temperature' parameters within the range -20°C to $+55^{\circ}\text{C}$. This temperature alert can therefore be used with both cold and warm produce.

Received temperature data outside the programmed parameters (typically 0°C - $+5^{\circ}\text{C}$) will activate the alert function and cause the display on the truck dashboard to flash on and off until the temperature returns within limits, when the unit will resume a steady display; or until the receiver is turned off. With this, the driver will alert with any changes and manage to take action when anything goes wrong with the temperature of their trailer.

2.9.4 RFID Pill Monitors Body Temperature [21]

Researchers in The Netherlands were able to monitor the body temperature by swallowed an RFID-based temperature sensor that measured their internal temperature. This will helped researchers identify potential health issues. This RFID technology has been used by volunteer participants at the world's largest marching event in the annual Four Days Marches of Nijmegen. With this pill, the researcher can monitor the marcher's core body temperatures so that they will not exceed recommended levels and prevent them from exhaustion or overheating.

Using complex event processing (CEP) technology, researchers were able to monitor and record the temperatures via a signal transmitted every ten seconds from the RFID "pill" to a receiving device in the volunteer's backpack. That data was then transmitted via Bluetooth to a GPS-enabled mobile phone to the operations center handled by the researcher themselves.

CEP is an event processing technology that allows an application to analyze multiple streams of event data, and then react to those conditions quickly. The CEP platform processed and analyzed the temperature data in real time. If a volunteer's body temperature was too high, officials could alert them to either rest or rehydrate using SMS text messaging, calling them on the mobile phone, or by alerting the onsite medical team to take action if needed.

CHAPTER 3

METHODOLOGY

3.1 Procedure Identification

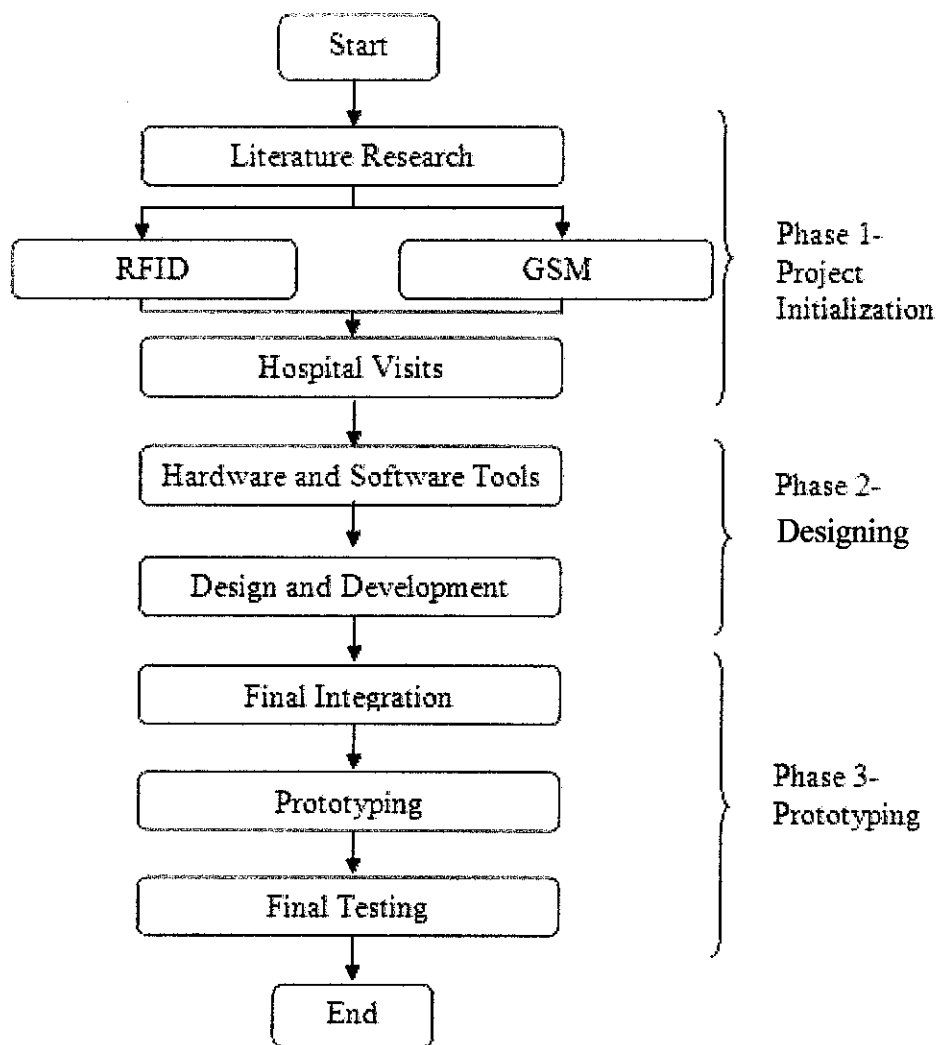


Figure 7 : Flow Chart

3.1.1 Phase one (Project Initialization)

Phase 1 is the early stage of the project development. Firm planning and understanding of the project is a must so that the project will be smoothly done. The understanding of the project is including the research on the technology that wants to be used that are RFID and GSM network.

Target hospital that will be implemented for this prototype is Hospital Sungai Buloh. The pioneer of this project; Mohamad Syukri Effendy has done this visit. The visit has given him the opportunity to meet the director of Hospital Sungai Buloh for professional opinion about the project and lots of advice given to see the project succeeded. This eventually will benefit the author as the successor for the project.

3.1.2 Phase two (Designing)

Phase 2 is where the designing the layout and the process of the system will be done. Choosing the right hardware to be used and development of the software and the database must be ready within this phase. Researchers have been made for the hardware and software needed for the project. The hardware needed are RFID Active Tag Temperature Sensor, RFID reader, and GSM Modem.

3.1.3 Phase Three (Prototyping)

Last but not least is phase 3, the beginning of the fabrication for the prototype. The integration will be focusing on the interfacing between devices, hardware and system of the project. In final testing stage, the project should been tested for its completion after the prototype finished with the combining of all of the hardware and the software application. The final testing will be made by replacing actual Malaria and Typhoid fever patient with a device that can control temperature. This stage is to make sure that the system functions well before the actual testing will be done at Hospital Sungai Buloh.

3.2 Project Overview

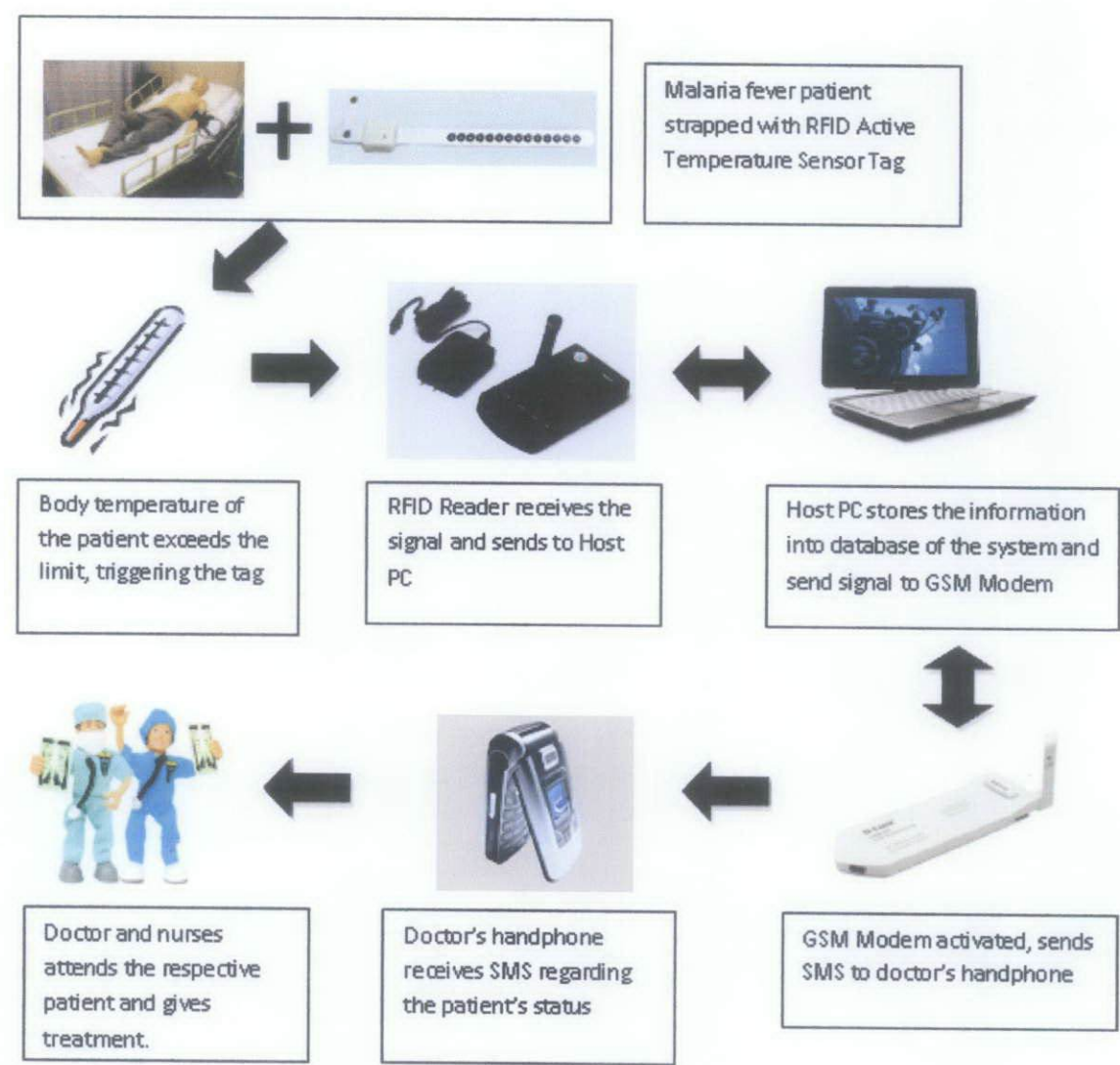


Figure 8: Project Overview

Patient will be strapped with the tag at his/her armpit to determine the body temperature. The limit for the temperature to exceed is 39°C. Once the system triggered, the tag will send signal to the reader attached with the Host PC. Then Host PC will trigger alarm on it for nurses to be alert and send signal to GSM Modem for it to be activated. SMS will be sent to doctor's hand phone for him to be notified regarding the patient's status. The patient will be then receiving treatment from both of the nurses and doctor.

3.3 RFID Hardware and Specification

For this project, there are two most important things needed. They are RFID Active Tag Temperature Sensor and RFID reader.

Although active RFID need a power source compared to passive and semi-passive RFID tags, it can store much bigger data and information. It also can broadcast response signal to the reader. This active tag is typically more reliable as there are only fewer errors than passive tags due to the ability for it to conduct 'session' with the reader. The Active Tag Temperature Sensor used for this project has characteristics as below;

- i. Long communication range
- ii. Having the capability of initiating communications (e.g. communicate with the reader each time patient having more than 39°C)
- iii. Having the capability to perform independent monitoring and control (e.g. monitor temperature of the patient each 4 hours)
- iv. Need battery power to function, which limit the lifetime and affect the cost for long-term maintenance.
- v. The tag can stand a harsh environment
- vi. Capable of perform independent monitoring and control

Second thing needed is RFID reader that used to interrogate an RFID tag. Readers communicate to the Host computer via an RS-232 cable or via an Ethernet network connection. Readers are used to read the tags and transmit the received data to the Host computer. Readers also used to enable, disable, wake up and program the tags. It has an antenna that emits radio waves while the tag responds by sending back its data. As in Figure 9, we can see that the RFID reader will use two frequencies that are 916.5 MHz and 433 MHz for transmitting and receiving signal. This is to avoid clashing of transmitting and receiving data from the tags. RFID reader for this project will be in beacon mode. Beacon mode is where the reader allowed the tag to continuously send the data to them in certain period minutes and place into the database.

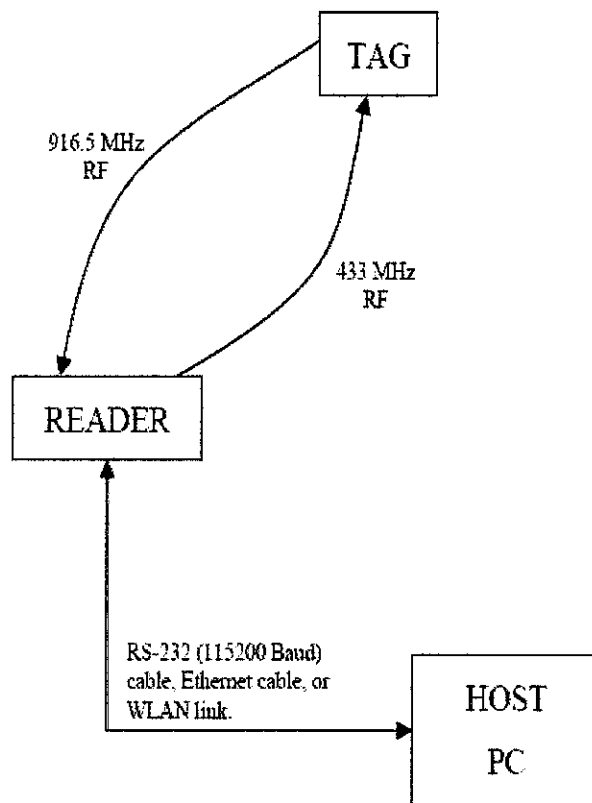


Figure 9: System flow for ActiveWave hardware used

Active Wave RFID reader field strength can be configured by ranging from 0 to 20. So, we can estimate the RF field from setting the strength ratio as shown in table 5 below.

Table 5: Setting ActiveWave Reader field strength ratio corresponds to estimated distance

Field Strength Ratio	Estimated Distance (m)	Field Strength Ratio	Estimated Distance (m)
1	4.25	11	46.75
2	8.50	12	51.00
3	12.75	13	55.25
4	17.00	14	59.50
5	21.25	15	63.75
6	25.50	16	68.00
7	29.75	17	72.25
8	34.00	18	76.50
9	38.25	19	80.75
10	42.50	20	85.00



Figure 10 : Active Wave RFID Kit

Figure 10 shows the hardware tools that will be used for this project; manufactured by Active Wave.

RFID Kit Components:

- I. 1 Active Wave Reader and Power Supply
- II. 1 Active Wave RJ-45 Cable
- III. 6 Active Wave Wristband Tags with Temperature Sensor
- IV. 1 Programming Station Software Application

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Interface

4.1.1 Interface for System Login and Main Menu

The first step, user such as the nurse or doctors needs to login their username and password as in Figure 11 below. After the users succeed with the 'Login', 'Main Menu' will come up as Figure 12. 'Start' button must be clicked to start the program and switch the program on. Then, the user can choose to click either 'System Administration' to register a new user of the system or new patient, or 'Patient Record' to view specific patient with their temperature body.

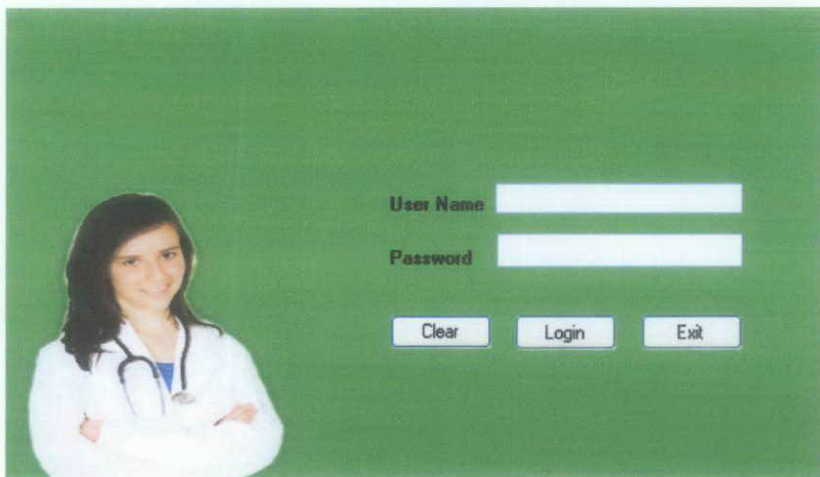


Figure 11: Login Menu

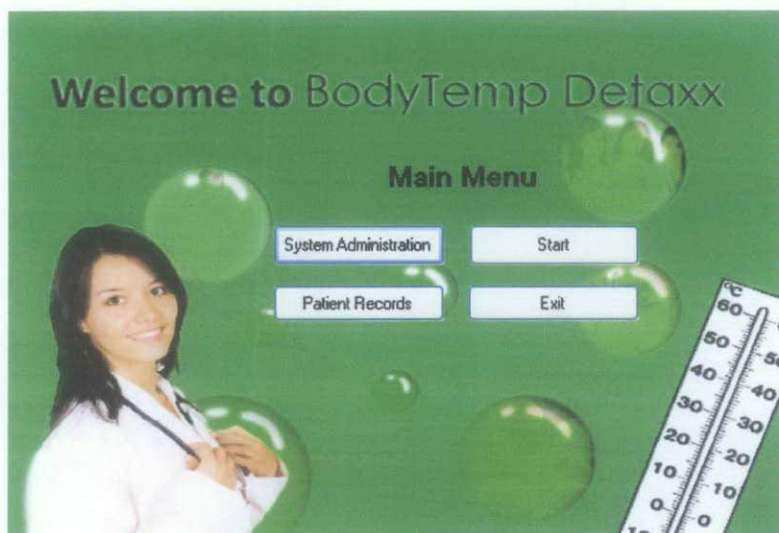


Figure 12: Main Menu

4.1.2 Interface for Administration

For administrator to register new patient or new user for the system, they just have to click ‘System Administration’ in Main Menu. This can be seen as in Figure 13. If they click on ‘Add User’, interface as in Figure 14 will come out while Figure 15 will come out if they click ‘Add Patient’. They just need to complete the form and click ‘Register’ and then all of the information will be kept in database of the system.

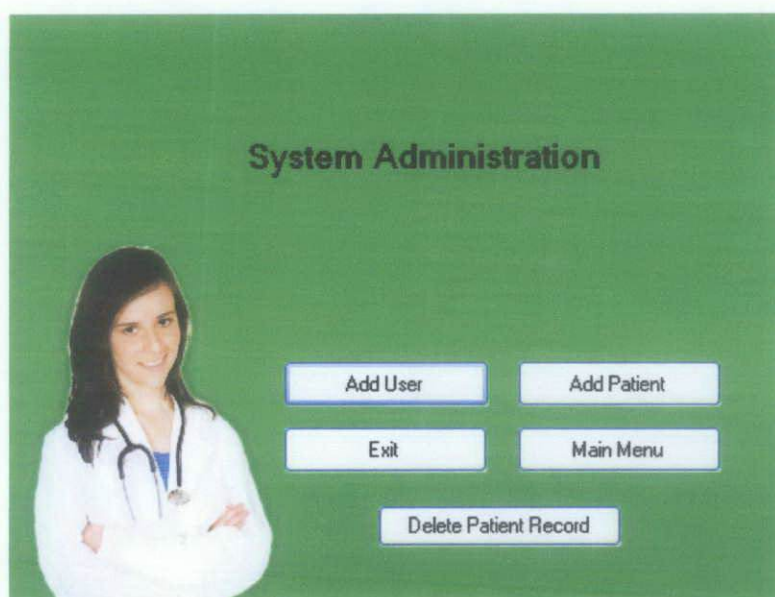
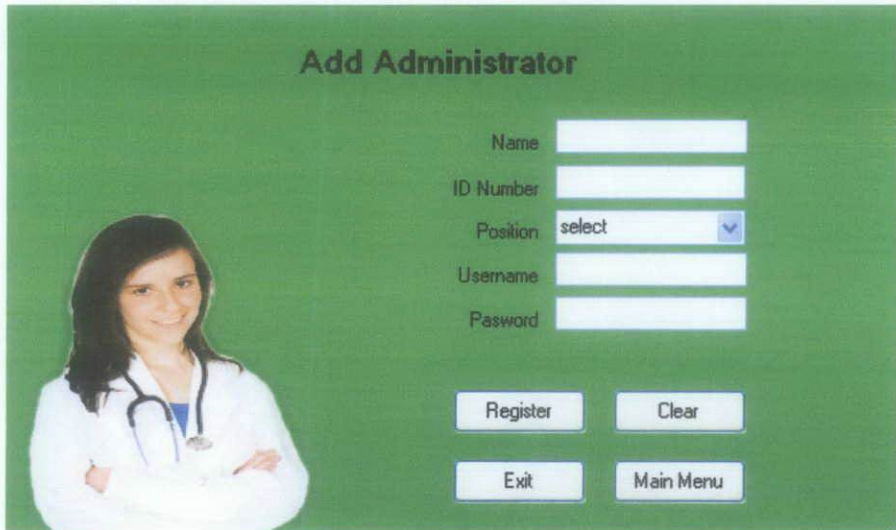


Figure 13: Administration Menu

The image shows a software interface for adding a new administrator. On the left, there is a photograph of a female doctor with dark hair, wearing a white lab coat and a stethoscope, with her arms crossed. The background is a solid green color. To the right of the photo, the title "Add Administrator" is centered at the top. Below the title, there are five input fields: "Name", "ID Number", "Position" (a dropdown menu with "select" and a downward arrow), "Username", and "Pasword" (note the spelling). At the bottom right, there are four buttons: "Register", "Clear", "Exit", and "Main Menu".

Add Administrator

Name

ID Number

Position

Username

Pasword

Figure 14: New User Registration Menu

The image shows a software interface for patient registration. On the left, there is a photograph of the same female doctor from Figure 14, wearing a white lab coat and a stethoscope, with her arms crossed. The background is a solid green color. To the right of the photo, the title "Patient Registration" is centered at the top. Below the title, there are seven input fields: "Patient Name", "I.C. Number", "Diagnosis", "Date of Birth", "Temperature", "Tag ID" (a dropdown menu with a downward arrow), and "Age". At the bottom, there are four buttons: "Register", "Clear", "Exit", and "Main Menu".

Patient Registration

Patient Name

I.C. Number

Diagnosis

Date of Birth

Temperature

Tag ID

Age

Figure 15: New Patient Registration Menu

If the doctors or admin want to register a new patient and using a tag ID that has been used before, they need to delete previous patient's information that use corresponding tag ID before proceed to register the new patient. This can be done by clicking 'Delete Patient Record' as in Figure 13 and enter corresponding tag ID that want to delete. The previous patient's information will be moved to archive database and corresponding tag ID can be use to register the new patient.

Figure 15 shows the patient registration page for ‘Add Patient’ section from previous administration page. The information required will include the patient’s name, i/c no, diagnosis of disease, date of birth, temperature on admission, patient tag ID that and age. The submit section is where the information will be stored at the database of the system as shown in Figure 16.

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

	name	ic_num	tagid_patid	dateofbirth	dateofadmin	diagnosis	temperature	timeofadmin	age
<input type="checkbox"/>	Aurangzeb	880828-05-2859	204	1988-08-28 23:03:58	2009-06-01 23:04:13	malaria	35	2009-06-01 23:04:37	17
<input type="checkbox"/>	Abu b. Ali	800101-23-4059	201	1980-01-01 12:23:23	2009-06-04 12:24:50	malaria	38	2009-06-04 12:24:50	24

Check All / Uncheck All With selected

Show : 30 row(s) starting from record # 0

Figure 16: Database for New Patient Registration section

4.1.3 Interface for Patient Record

When the nurse or doctor want to gather about their patient’s information, they just have to click on ‘Patient Record’ on the ‘Main Menu’ and figure as in Figure 17 will come out. From here, they can decide either they want to retrieve previous patient’s record or view current patient’s record. They just have to click either button as in Figure 17 and then enter patient ID and click ‘Confirm Details’ as in Figure 18. The patient’s information will be gathered from related database and came out for the doctor to analyze the trend of the patient’s temperature.

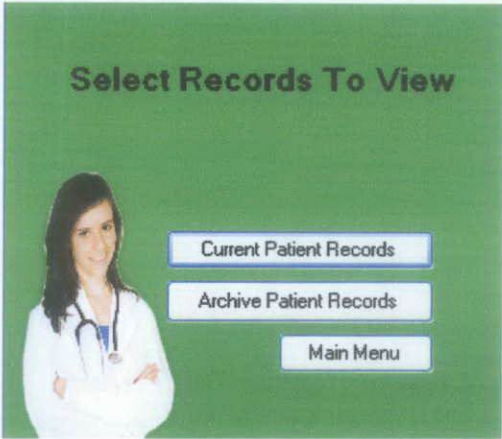


Figure 17: Patients Record

Patient Records

Enter patient ID

203

Search

Confirm Details

Patient Name

Abd. Karim bin Mohamad

Patient's Age

22

Diagnosis

Typhoid

Date of Birth

1985-03-14 00:00

Date of Admission

2009-09-14 14:15

IC Number

070107-43-5655



Temperature	Date	Time
39.25	2009-10-04	18:03:53
38.625	2009-10-04	18:02:51
38.625	2009-10-04	18:02:47
37.25	2009-10-04	18:01:20
37.25	2009-10-04	18:01:17
36	2009-10-04	18:00:18
36	2009-10-04	18:00:16
36	2009-10-04	18:00:15
36	2009-10-04	18:00:15
36	2009-10-04	18:00:14
34	2009-10-04	17:59:13
34	2009-10-04	17:59:11
34	2009-10-04	17:59:08
30.5	2009-10-04	17:57:13
34	2009-10-04	17:59:09
30.5	2009-10-04	17:57:12
30.5	2009-10-04	17:53:41
26	2009-10-01	13:58:32

Main Menu

Exit

Figure 18: Example of Patient Record

4.1.4 Interface between Database and GSM Modem

Whenever there is an alert about any patient who reached high body temperature, the popup as in Figure 19 below will come out. After the nurse notify with it and they click ‘Ok’ button, they need to click another icon at the desktop named ‘SMS Alert’. The interface for this ‘SMS Alert’ is as in Figure 20. The nurse just has to fill the doctor’s ID no and patient’s tag no as indicate in popup (Figure 19) and click ‘Check’. Once they did that, the interface will interact with the database and fill in the doctor’s name, doctor’s telephone number and patient’s information and send the message once the nurse click ‘SMS Doc’.

203

temperature very high

OK

Figure 19: Alert Form

SMS Alert

Dr. ID Dr. Name Dr. HandPhone

ATTENTION The Patient: Abd. Karim bin Mohamad Age 22 Id Number 870107-43-5655 Admitted on 9/14/2009 2:15:34 PM Diagnosed with Typhoid Temperature is Critical

Patient ID

Figure 20: SMS Alert Form

4.2 Demo in front of Sg. Buloh Hospital's Director.

On 17th of September, the author and supervisor of this project, Ms. Hanita have made an appointment with Director of Hospital Sg. Buloh, Dr. Khalid Ibrahim. This appointment is made to have the first demo of the system and get feedback from him. Few feedbacks and information have been clarified further by him especially about the need of the system for Malaria and Typhoid fever patients. As being stated by Dr. Khalid, the bacterium of the fever is high when the body temperature is at its peak (39-40°C), and that is the time to take blood sample to check whether there are active parasite or bacterium that related to malaria or typhoid. So, this system has eased the doctor and nurse from checking the pattern of the patient's body temperature manually to have the diagnosis of the fever.



Figure 21: Author with Project Supervisor at Hospital Sg. Buloh

4.3 Testing the Device

4.3.1 *Range between each tag and reader*

This test needs to be done to check if any interference happened between each tag and reader that affects the efficiency of the system to send and receive the signal. If interference happened, it can defect the system where maybe the tag cannot send the signal to the reader although it has been triggered by the temperature of patient's body. The distance range from reader and the tag must also be tested to check for their capabilities of communicating with each other although at a long distance.

4.3.2 *User friendly interface*

The system has been tested and demonstrated in front of few people just to get their opinion about the interface and adaptability of the system. It must be easy to use and user friendly.

4.3.3 *Error percentage of the system*

After done with lab test and hospital test for the system, it must be check for its error percentage. This is vital to reconfigure the system to have the right configuration time so that the alert popup when the core body's temperature of the fever patient is at its peak. The formula to check the error percentage is as below;

$$\text{Error \%} = \left[\frac{\text{Theoretical temperature} - \text{Real temperature reading}}{\text{Theoretical temperature}} \right] \times 100$$

4.4 Lab Test

The objectives of lab testing are to show that the system is working in the range of fever temperature (within 36 to 40°C) and also to check for tag's sensitivity, accuracy and performance. For the test, we have used temperature controller which is magnetic stirrer to control the tag's temperature. Below is the result for the lab test;

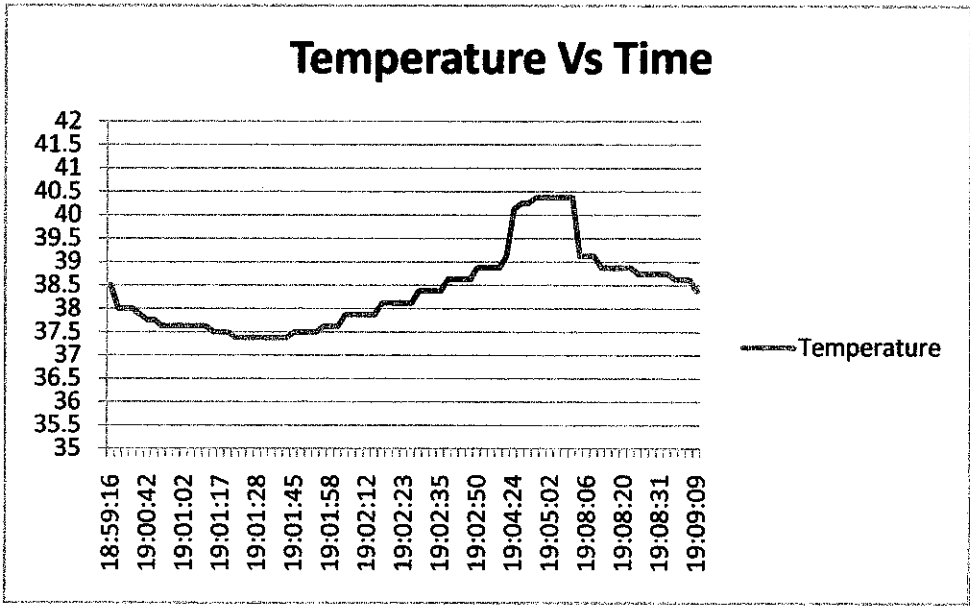


Figure 22: Temperature vs. time for lab testing

As we can see in the graph above, the system is perfectly working in the temperature range needed to check for the fever patients. Besides, the system is quite sensitive to any changes of the temperature where it can detect the changes within seconds.

4.5 Hospital Tests

Hospital testing has been done with objectives of checking the accuracy, reliability, radius of the system’s RFID reader and tag and also check for any error of the system. This test has been done on 23rd and 24th of November 2009 and involved two patients from Hospital Sg. Buloh.

4.5.1 Patient 1

For first testing, we have an Indonesian patient named Adnan Rustandik, age 19. Result for the test on patient 1 is as below;

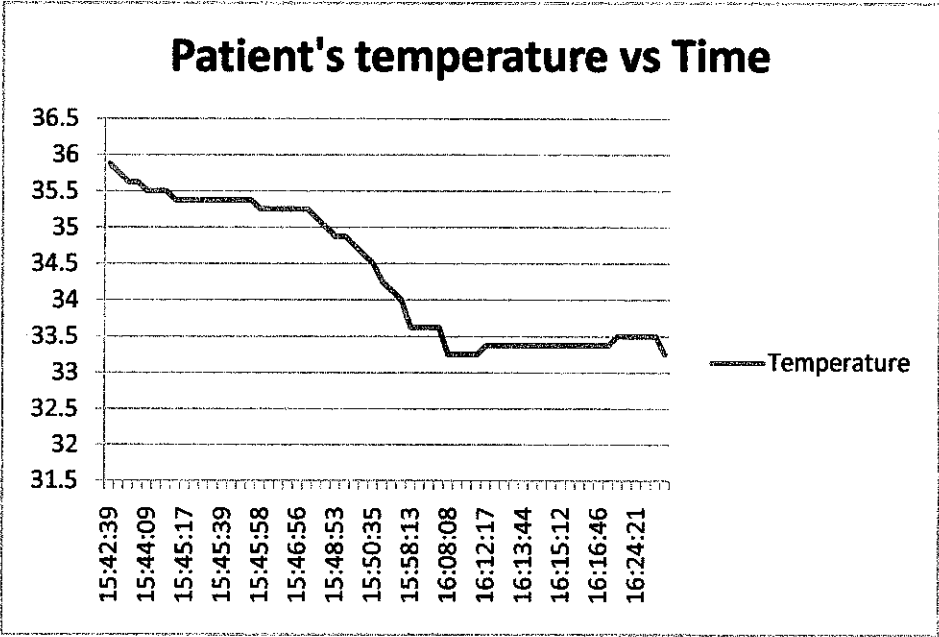


Figure 23: Patient’s 1 Temperature vs. Time

As shown in the figure above, the patient has quite a high fever but after 5 minutes, his body temperature decreased. This is due to his doctor’s action to inject intravenous (iv) fluid into his body. After iv fluid fully inserted into the patient’s body, his temperature become stable around 33.5°C.

4.5.2 Patient 2

For first testing, we have an Indonesian patient named Hary Urtiadi, age 31. His body's temperature is as shown as figure below.

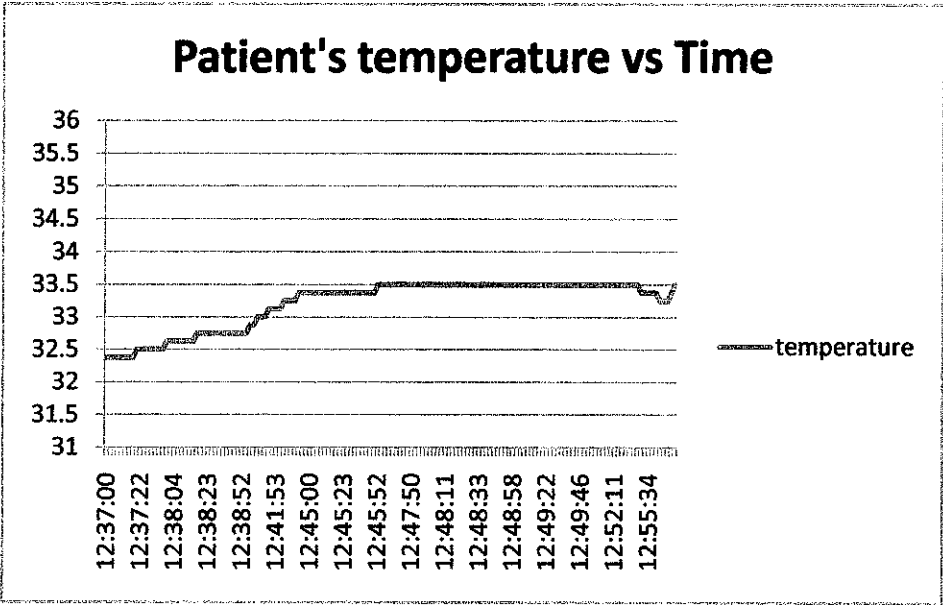


Figure 24: Patient's 2 Temperature vs. Time

As shown in the figure above, although the patient is suspected with Malaria fever, his body temperature is already stable due the fact that his doctor has already treated him with appropriate medicine. So, he has a stable temperature around 33.5°C. The temperature is lower than average at the first three minutes just because of the ambient room temperature that is quite low due to the air-conditioner in the ward. After the tag getting used with the patient's armpit, the tag's temperature shows according to the core body temperature of the patient.

4.6 Test Result

After done with hospital tests and lab test, the system has been proved that it can cover around five to seven meters radius from the RFID reader and its tag. From the data gathered, we know that there are errors in measuring patient's body temperature. The factor that effects the measurement depends on where the tag is placed on the body of the patient. As in this case, the tag is being put at the armpit of the patient. Normal temperature in the mouth is around 37°C while normal temperature on the armpit is around 36°C. So, there is one degree Celsius different between normal temperatures in the mouth and on the armpit.

The average temperature taken from the hospital tests is around 33.5°C. This average can be taken as a normal temperature of the core body temperature of a patient. So, from data that we have, the error for this system can be calculated as below;

$$\begin{aligned}\text{Error \%} &= \left[\frac{\text{Theoretical temperature} - \text{Real temperature reading}}{\text{Theoretical temperature}} \right] \times 100 \\ &= \left[\frac{36^\circ\text{C} - 33.5^\circ\text{C}}{36^\circ\text{C}} \right] \times 100\% \\ &= 6.94\%\end{aligned}$$

The error is around 7%. So, we need to reconfigure the system to have the threshold value to have 7% less than its theoretical value. As for the system, if the peak threshold of core body temperature on the armpit of a fever patient is around 38°C, we need to change it around 35.5 °C for the alert to popup.

In order to improve the test result, the system needs to be tested on more patients. With a lot more tests, the system can have more accurate error percentage by analyzing and calculate from all data gathered.

4.7 Challenges faced

4.7.1 Building a source code for the system

This system has a quite complicated source code. This becomes a challenge to the author as the author only having the basic knowledge of the VB language. The author has to do a lot of research about VB and ask for help and support from VB users. Below is the flow diagram of the system before the source code is built accordingly.

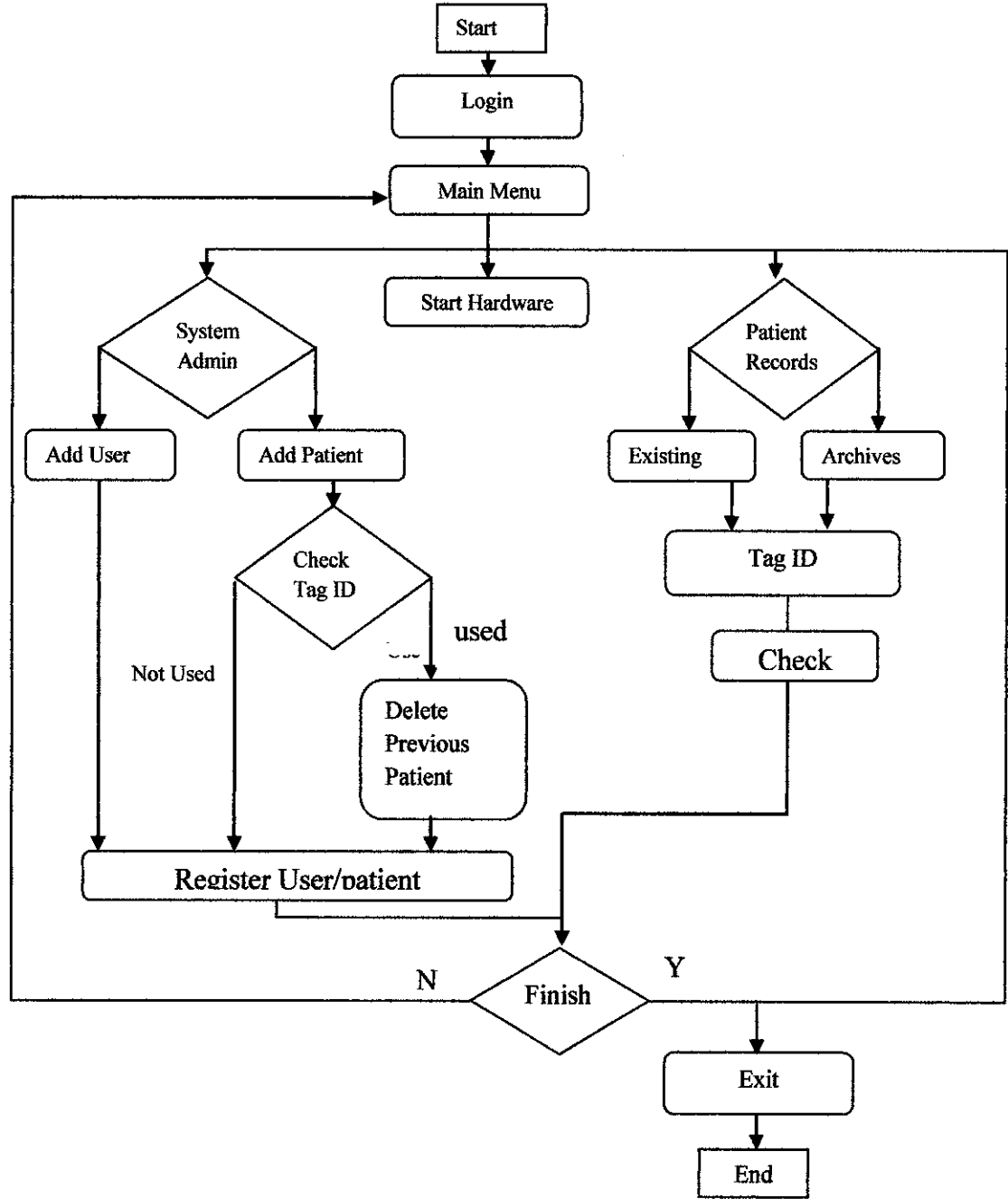


Figure 25: Flow diagram of the system

4.7.2 Building the database for the system

One of the other challenges for the author to build up this system is to construct database system. Databases for this system is quite complicated where there are five databases shown as below;

- Temperature database
- Admin database
- Current patient database
- Archive patient database
- Archive database

Archive database is needed to connect the entire related patient's name with the corresponding patient's tag ID. It is important to have a correct connection of specific tag ID with specific patient's info to retrieve archive database.

All of these databases are important to make sure this system can run smoothly. A lot of try error and research has been done to make sure that the database is complete with all of the data needed for the system to functioning well. Databases of the system are shown as in Appendix A.

4.7.3 Integration between RFID hardware and GSM modem

In the first planning, GSM modem is part of the system where whenever the patient's core body is above 39°C, the system will trigger the GSM modem to straight away send the SMS the doctor who is on duty. The problem that happened is that GSM and RFID cannot be integrated together trough Visual Basic (VB). The solution that has been done is that the system for temperature tracking is separated with the SMS alert although both of the system used the same database.

Now, whenever the alert popup from the temperature tracking system, the nurses have to click 'SMS Alert' icon and insert patient's tag ID and ID for the doctor who is in charge at that moment before they can send the SMS.

4.7.4 Hardware Reliability

Active Wave RFID kit having the reliability issues such as battery of the tags only last for about two to three months and the distance range between the tags and reader is quite low than what it supposed to be. Issue of the battery has increased the cost of the system in a long term especially in maintenance cost.

The solution that applicable is the tag will take measurement of patient's body temperature once every four hours as suggested by doctor and not by every few seconds as this prototype has made it. While regarding the distance range covered by single reader, the placement of it needs to be taking into account when implementing the system. As in this case, the reader must be put at a higher place and at the centre of the room so that the radius for the reader to detect any tags surrounding it will be increased.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

As far as this project concern, we can conclude that for the lab test, result shows that the system can perfectly working in range of temperature needed (36-40°C) and sensitive to temperature changes while result for hospital tests shows that there is an error in measurement around 7%, but the system need to be tested on more patient to make the error percentage to be more accurate.

Besides, we also can conclude that this system is essential for the doctors and nurses in detecting and serve the Typhoid and Malaria patient well where it can measure and monitor the body temperature of the patient. The system can alert the nurse at their station and notify via SMS to the doctor definitely a good feature that can reduce the possibilities of fatal that can happen to the patient.

5.2 Recommendations

There are few of recommendations that can be done to improve this system better. They are as follows;

- The system can also be done to other type of fever or any temperature related disease such as dengue and cikunguya fever. This will benefit more people in wider cases of fever.
- Implementation of other RFID sensor than not only detects body temperature but also other necessary information such as blood pressure. This absolutely helps the doctor and nurses to act faster and more efficient.

REFERENCES

- [1] Online Encyclopedia. *Online encyclopedia dedicated to computer technology*. [Online] [Cited: February 3, 2009.] <http://www.webopedia.com/TERM/R/RFID.html>.
- [2] World Health Organization. *World Health Organization Web site*. [Online] [Cited: February 3, 2009.] <http://www.who.int/topics/malaria/en/>.
- [3] MedecineNet.com. *MedecineNet.com websites*. [Online] [Cited: February 3, 2009.] <http://www.medicinenet.com/malaria/article.htm>.
- [4] World Health Organization. *Management of Severe Malaria*. Geneva : World Health Organization, 2000. ISBN.
- [5] Ministry of Health Malaysia. *Ministry of Health Malaysia Web site*. [Online] [Cited: February 3, 2009.] moh.gov.my/MohPortal/health_fact.jsp.
- [6] Wellsphere. *Wellsphere Websites*. [Online] Sept 8, 2008. [Cited: February 4, 2009.] <http://stanford.wellsphere.com/wellguide.s?articleId=380§ionId=1&searchString=Causes+Of+Fever>.
- [7] USA.gov. *Department of Health and Human Services web site*. [Online] [Cited: February 4, 2009.] <http://www.cdc.gov/ncidod/dbmd/diseaseinfo/default.htm>.
- [8] RFID Journal Staff Writer. "RFID:Frequently Asked Questions", 2005. RFID Journal. <http://www.rfidjournal.com/faq>
- [9] Craig Ross, Ricardo Goto. RFID Proximity Security System.
- [10] Harold G. Clampitt, M., PE(2007) "RFID Certification Textbook", 509 pages.
- [11] Buzzle.com. *Buzzle.com websites. RFID technology* [Online] 2000-2008. [Cited: October 29, 2009.] <http://www.buzzle.com/articles/passive-vs-active-rfid-tags.html>

- [12] Linda Castro & Samuel Fosso Wamba, "An Inside Look at RFID Technology", *Journal of Technology Management & Innovation*, 2007, Volume 2 Issue 1, page 128.
- [13] "Learn More About Radio Frequency Identification "
<http://www.impinj.com/about-rfid/default.aspx>. [Online] [Cited: November 20, 2009]
- [14] RFID-Handbook. *Types of RFID*. [Online] Sept 26, 2009. [Cited: October 18, 2009.] http://rfid-handbook.de/rfid/types_of_rfid.html
- [15] Klaus Finkenzeller. "RFID Handbook - Fundamentals and Application in Contactless Smart Cards and Identification, Second Edition." Wiley Publishing, Inc.
- [16] Grassfstra, A. "RIFD Toys. Cool Projects for Home, Office, and Entertainment." Wiley Publishing, Inc.
- [17] Joachim Tisal, "The GSM Network GPRS Evolution: One Step Towards UMT", Chichester, England: John Wiley & Sons, 2001.
- [18] UsingRFID.com. *UsingRFID.com Web site*. [Online] Digital Angle Corporation. [Cited: February 8, 2009.]
<http://www.usingrfid.com/news/read.asp?lc=q95777tx597zq>.
- [19] RFID Convocation. *RFID Convocation Web site*. [Online] [Cited: February 08, 2009.] www.rfidconvocation.eu/Papers%20not%20presented/RFIDSmartTemperatureSensor.pdf
- [20] Securetemp. *Securetemp Web site*. [Online] Securetemp. [Cited: February 08, 2009.] <http://www.secureseal.com/images/pdfs/securetemp.pdf>.
- [21] RFID update - The RFID Industry Daily. *RFID update - The RFID Industry Daily Web site*. [Online] [Cited: March 3, 2009.]
<http://www.rfidupdate.com/news/07292008.html>.

phpMyAdmin

Server: localhost Database: hospital_db Table: temptrack

Database

hospital_db (5)

hospital_db (5)

archives

archive_names

patient_details

registration

temptrack

Field

Type

Collation

Attributes

Null

Default

Extra

Action

☐

tagID

varchar(6)

latin1_general_ci

No

☐

name

varchar(30)

latin1_general_ci

No

☐

temperature

varchar(6)

latin1_general_ci

No

☐

time

datetime

No

0001-01-01 12:00:00

Check All / Uncheck All

With selected:

Print view

Propose table structure

Add 1 field(s)

At End of Table

At Beginning of Table

After tagID

Go

Show: 30 row(s) starting from record # 30

in horizontal mode and repeat headers after 100 cells

	tagID	name	temperature	time
<input type="checkbox"/>	203	True	34.625	2009-11-30 17:12:33
<input type="checkbox"/>	201	True	25.375	2009-11-30 17:12:19
<input type="checkbox"/>	203	True	34.625	2009-11-30 17:12:12
<input type="checkbox"/>	201	True	25.375	2009-11-30 17:11:58
<input type="checkbox"/>	203	True	34.625	2009-11-30 17:11:48
<input type="checkbox"/>	203	True	34.625	2009-11-30 17:11:16
<input type="checkbox"/>	203	True	34.5	2009-11-30 17:10:51
<input type="checkbox"/>	203	True	34.5	2009-11-30 17:10:29
<input type="checkbox"/>	203	True	34.5	2009-11-30 17:10:08
<input type="checkbox"/>	203	True	34.375	2009-11-30 17:09:47
<input type="checkbox"/>	203	True	34.375	2009-11-30 17:09:19
<input type="checkbox"/>	203	True	34.375	2009-11-30 17:08:58
<input type="checkbox"/>	203	True	34.375	2009-11-30 17:08:37
<input type="checkbox"/>	201	True	26.75	2009-11-30 17:08:10
<input type="checkbox"/>	203	True	34.375	2009-11-30 17:08:08
<input type="checkbox"/>	203	True	34.375	2009-11-30 17:07:47
<input type="checkbox"/>	201	True	27.125	2009-11-30 17:07:38

phpMyAdmin

Server: localhost Database: hospital_db Table: registration

Database

hospital_db (5)

hospital_db (5)

archives

archive_names

patient_details

registration

temptrack

Field

Type

Collation

Attributes

Null

Default

Extra

Action

☐

ID

varchar(30)

latin1_general_ci

No

☐

Post

varchar(30)

latin1_general_ci

No

☐

user_name

varchar(30)

latin1_general_ci

No

☐

password

varchar(30)

latin1_general_ci

No

☐

name

varchar(30)

latin1_general_ci

No

☐

DoctorHP

varchar(15)

latin1_general_ci

Yes

000 0000000

Check All / Uncheck All

With selected:

Print view

Propose table structure

Add 1 field(s)

At End of Table

At Beginning of Table

After ID

Go

Query results: operations

Print view

Print view (with full texts)

Export

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

Go

	ID	Post	user_name	password	name	DoctorHP
<input type="checkbox"/>		doctor		123		000 0000000
<input type="checkbox"/>	doctor1	Doctor	dr. shariffudin	123	shariffudin	0195262206

Check All / Uncheck All

With selected:

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

phpMyAdmin Server: localhost Database: hospital_db Table: patint_details

Database: hospital_db (5)

Field Type Collation Attributes Null Default Extra Action

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> name	varchar(50)	latin1_general_ci		No			
<input type="checkbox"/> ic_num	varchar(15)	latin1_general_ci		No			
<input type="checkbox"/> tagid_patid	varchar(15)	latin1_general_ci		No			
<input type="checkbox"/> dateofbirth	datetime			No			
<input type="checkbox"/> dateofadmin	datetime			No			
<input type="checkbox"/> diagnosis	varchar(20)	latin1_general_ci		No			
<input type="checkbox"/> temperature	int(11)			No			
<input type="checkbox"/> age	int(11)			No			

Check All / Uncheck All With selected:

Print view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After name Go

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

	name	ic_num	tagid_patid	dateofbirth	dateofadmin	diagnosis	temperature	age
<input type="checkbox"/>	Tadiwa	12341324	200	1990-12-12 00:00:00	2009-11-22 15:54:19	Malaria	34	23
<input type="checkbox"/>	ili	1234	201	2009-12-24 00:00:00	2009-11-24 12:50:21	distance testing	35	2
<input type="checkbox"/>	Hary Uriadi	indon1234	203	1976-08-15 00:00:00	2009-11-24 12:38:32	Malaria	37	31
<input type="checkbox"/>	pudin	12345	205	1990-11-23 00:00:00	2009-11-23 15:50:04	tstng distance	37	19

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

phpMyAdmin Server: localhost Database: hospital_db Table: archive_names

Database: hospital_db (5)

Field Type Collation Attributes Null Default Extra Action

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> name	varchar(85)	latin1_general_ci		Yes	NULL		
<input type="checkbox"/> ic_num	varchar(15)	latin1_general_ci		Yes	NULL		
<input type="checkbox"/> tagid	int(11)			Yes	NULL		
<input type="checkbox"/> dateofbirth	datetime			Yes	NULL		
<input type="checkbox"/> dateofadmin	datetime			Yes	NULL		
<input type="checkbox"/> diagnosis	varchar(45)	latin1_general_ci		Yes	NULL		
<input type="checkbox"/> temperature	int(11)			Yes	NULL		
<input type="checkbox"/> age	int(11)			Yes	NULL		
<input type="checkbox"/> count	int(11)			No		auto_increment	

Check All / Uncheck All With selected:

Print view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After name Go

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

	name	ic_num	tagid	dateofbirth	dateofadmin	diagnosis	temperature	age	count
<input type="checkbox"/>	Tadiwa	12341324	200	1990-12-12 00:00:00	2009-11-22 15:54:19	Malaria	34	23	1
<input type="checkbox"/>	john	0128390821	201	1990-12-12 00:00:00	2009-11-23 12:38:26	malaria	37	19	2
<input type="checkbox"/>	Adnan Rustandik	indon123	203	1990-11-23 00:00:00	2009-11-23 15:43:39	denggi	37	19	3
<input type="checkbox"/>	Adnan Rustandik 2	indon12345	201	1990-11-23 00:00:00	2009-11-23 15:44:52	denggi	37	19	4
<input type="checkbox"/>	pudin	12345	205	1990-11-23 00:00:00	2009-11-23 15:50:04	tstng distance	37	19	5
<input type="checkbox"/>	temp testing	091123-43-5355	201	2009-11-23 00:00:00	2009-11-23 18:48:45	malaria	37	1	7
<input type="checkbox"/>	temp testing 2	1234	203	2009-11-23 00:00:00	2009-11-23 19:02:20	typhoid	37	1	8
<input type="checkbox"/>	Hary Uriadi	indon1234	203	1976-08-15 00:00:00	2009-11-24 12:38:32	Malaria	37	31	9
<input type="checkbox"/>	ili	1234	201	2009-12-24 00:00:00	2009-11-24 12:50:21	distance testing	35	2	12

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

php

Server: localhost Database: hospital_db Table: archives

Structure

SQL

Search

Insert

Export

Import

Operations

Empty

Drop

Database

hospital_db (5)

hospital_db (5)





























































archives

archive_names

patient_details

registration

temprock

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> name	varchar(65)	latin1_general_ci		Yes	NULL		     
<input type="checkbox"/> lc_num	varchar(25)	latin1_general_ci		Yes	NULL		     
<input type="checkbox"/> tagid	varchar(15)	latin1_general_ci		Yes	NULL		     
<input type="checkbox"/> dateofbirth	datetime			Yes	NULL		     
<input type="checkbox"/> dateofadmin	datetime			Yes	NULL		     
<input type="checkbox"/> diagnosis	varchar(45)	latin1_general_ci		Yes	NULL		     
<input type="checkbox"/> temperature	varchar(6)	latin1_general_ci		Yes	NULL		     
<input type="checkbox"/> age	int(11)			Yes	NULL		     
<input type="checkbox"/> time	datetime			Yes	NULL		     
<input type="checkbox"/> count	int(11)			No		auto_increment	     

Check All / Uncheck All With selected

Print view

Propose table structure

Add 1 field(s)

At End of Table

At Beginning of Table

After name

Go

Show: 30 row(s) starting from record # 30

>

>>







































Page number: 1

in horizontal

mode and repeat headers after 100 cells

Go

Sort by key: None

	name	lc_num	tagid	dateofbirth	dateofadmin	diagnosis	temperature	age	time
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		39.25	0	2009-11-23 12:49:57
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		39.375	0	2009-11-23 12:50:32
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		39.25	0	2009-11-23 12:51:38
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.375	0	2009-11-23 12:53:05
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.25	0	2009-11-23 12:53:26
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.5	0	2009-11-23 12:53:55
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.5	0	2009-11-23 12:54:22
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.375	0	2009-11-23 12:54:44
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.125	0	2009-11-23 12:55:07
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.125	0	2009-11-23 12:55:28
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31	0	2009-11-23 12:55:49
<input type="checkbox"/>  			203	0000-00-00 00:00:00	0000-00-00 00:00:00		35.875	0	2009-11-23 15:42:39
<input type="checkbox"/>  			203	0000-00-00 00:00:00	0000-00-00 00:00:00		35.875	0	2009-11-23 15:42:39
<input type="checkbox"/>  			203	0000-00-00 00:00:00	0000-00-00 00:00:00		35.875	0	2009-11-23 15:42:39
<input type="checkbox"/>  			201	0000-00-00 00:00:00	0000-00-00 00:00:00		31.75	0	2009-11-23 15:42:40
<input type="checkbox"/>  			203	0000-00-00 00:00:00	0000-00-00 00:00:00		35.875	0	2009-11-23 15:42:41
<input type="checkbox"/>  			203	0000-00-00 00:00:00	0000-00-00 00:00:00		35.75	0	2009-11-23 15:42:54
<input type="checkbox"/>  			203	0000-00-00 00:00:00	0000-00-00 00:00:00		35.75	0	2009-11-23 15:42:55
<input type="checkbox"/>  			203	0000-00-00 00:00:00	0000-00-00 00:00:00		35.75	0	2009-11-23 15:43:01

i. Login

```
Imports System.IO
Imports MySql.Data.MySqlClient
```

```
Public Class login
```

```
    Private Sub login_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
    End Sub
```

```
    Private Function checkPswd(ByVal name As String, ByVal pswd As String) As
Boolean
```

```
        Dim found As Boolean = False
        Dim rows As Integer = 0
        Dim myConnString As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
        Dim cn As New MySqlConnection(myConnString)
        Dim cmdGH As New MySqlCommand("SELECT * FROM registration WHERE
user_name = '" & name & "'AND password = '" & pswd & "'", cn)
        Dim daGH As New MySqlDataAdapter
        Dim dsGH As New DataSet
        Dim dtGH As New DataTable

        cn.Open()

        Try
            daGH.SelectCommand = cmdGH
            daGH.Fill(dsGH, "employees")
            dtGH = dsGH.Tables("employees")
            rows = dtGH.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!!")
        End Try

        cn.Close()

        If (rows > 0) Then
            found = True
        Else
            found = False
        End If

        Return found
```

```
End Function
```

```
    Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLogin.Click
```

```

If (checkPswd(txtUname.Text, txtPswd.Text) = True) Then
    Main_Menu.Show()
    Me.Hide()
Else
    MessageBox.Show("Incorrect User name or Password")
End If
End Sub

```

```

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
    txtUname.Text = ""
    txtPswd.Text = ""
End Sub

```

```

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
    Me.Close()
End Sub

```

End Class

ii. Main menu

Public Class Main_Menu

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

```

```

    Dim drExit As DialogResult
    drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Close()
    End If
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Go to patient Records ?", "Patient Records", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Hide()
        records_menu.Show()
    End If
End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim drExit As DialogResult

```

```

        drExit = MessageBox.Show("go to Aministartion module ?", "Administration", _
            MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
            MessageBoxDefaultButton.Button2)
        If drExit = DialogResult.Yes Then
            Me.Hide()
            System_admin.Show()
        End If
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs)

    End Sub

    Private Sub Main_Menu_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles BtnConnectStart.Click
        ConnStart.Show()
    End Sub
End Class

```

iii. Start Connection

```

Imports System.Data
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class ConnStart

    Dim Hconn As IntPtr
    Dim readerIP(20) As Byte
    Dim readerPort As UInt16
    Dim commPort As UInt32
    Dim commBaud As UInt32
    Dim myPKTID As Integer
    Dim registered As Boolean
    Dim ActiveWaveAPI As AW_API_NET.APINetClass = New
        AW_API_NET.APINetClass
    Dim ReaderEventHandler As AW_API_NET.fReaderEvent
    Dim TagEventHandler As AW_API_NET.fTagEvent
    Dim pubReaderID As Integer

```

```
Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
```

```
Private Sub ConnStart_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
    readerPort = Convert.ToUInt16(10001)
    commPort = Convert.ToUInt32(1)
    commBaud = Convert.ToUInt32(115200)
```

```
    ' Initialize callback functions
    ReaderEventHandler = New AW_API_NET.fReaderEvent(AddressOf
Me.OnReaderEvent) 'AddressOf Me.OnReaderEvent
    TagEventHandler = New AW_API_NET.fTagEvent(AddressOf Me.OnTagEvent)

    registered = False
```

```
End Sub
```

```
Private Function OnReaderEvent(ByVal readerEvent As
AW_API_NET.rfReaderEvent_t) As Integer
```

```
    Dim ipStr As String = ""
```

```
    'Beep()
    If
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SCAN_
NETWORK)) Then
        For i As Integer = 0 To readerEvent.ip.Length - 1
            ipStr += Convert.ToChar(readerEvent.ip(i))
        Next i
        IPListBox.Items.Add(ipStr)
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_OPEN_
SOCKET)) Then
        For i As Integer = 0 To readerEvent.ip.Length - 1
            ipStr += Convert.ToChar(readerEvent.ip(i))
        Next i
        'AddMsg("Socket Opened IP = " + ipStr)
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_CLOSE
_SOCKET)) Then
        For i As Integer = 0 To readerEvent.ip.Length - 1
            ipStr += Convert.ToChar(readerEvent.ip(i))
        Next i
        'AddMsg("Socket Closed IP = " + ipStr)
```

```
    Dim index As Integer
    If ipStr.Length > 0 Then
        index = IPListBox.FindStringExact(ipStr)
        If index >= 0 Then
            IPListBox.Items.RemoveAt(index)
```

```

        End If
    End If
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_STD_F
GEN_POWERUP)) Then
        'FGenIDTextBox.Text = readerEvent.fGenerator.ToString()
        'AddMsg("STD FGen Powered UP")
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_READ
ER_POWERUP)) Then
        'ReaderIDTextBox.Text = readerEvent.reader.ToString()
        'tText(readerEvent.reader.ToString())
        'AddMsg("Reader Powered UP")
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_QUER
Y_STD_FGEN)) Then
        Dim str As String

        'AddMsg(readerEvent.eventType.ToString)
        'AddMsg(AW_API_NET.APIConsts.RF_READER_POWERUP.ToString)

        'AddMsg("STD FGEN Query _____")
        str = readerEvent.smartFgen.fsValue
        'AddMsg("FS Value = " + str)
        str = readerEvent.smartFgen.txTime
        'AddMsg("TX Time = " + str)
        str = readerEvent.smartFgen.waitTime
        'AddMsg("Wait Time = " + str)
        str = readerEvent.smartFgen.assignRdr
        'AddMsg("Assigned Rdr = " + str)
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_GET_R
DR_FS)) Then
        'FSTextBox.Text = readerEvent.smartFgen.fsValue.ToString()
        'AddMsg("Reader ID:" + readerEvent.reader.ToString() + " FS:" +
readerEvent.smartFgen.fsValue.ToString())
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SET_R
DR_FS)) Then
        'AddMsg("Reader FS was set successfully")
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SCAN_
IP)) Then
        ipStr = GetStringIP(readerEvent.ip)
        If ipStr.Length > 0 Then
            If IPListBox.FindStringExact(ipStr) = -1 Then
                IPListBox.Items.Add(ipStr)
            End If
            'AddMsg("ScanIP IP=" + ipStr)
        End If
    End If
End If

```

```

    ReportReaderEvent(readerEvent)

    Return 0
End Function

Private Sub ReportReaderEvent(ByRef readerEvent As
AW_API_NET.rfReaderEvent_t)

    If readerEvent.errorStatus.ToString() = 0 Then
        commStatus.Text = "ON"
        pubReaderID = readerEvent.reader.ToString()
    End If

End Sub

Private Function OnTagEvent(ByVal tagEvent As AW_API_NET.rfTagEvent_t) As
Integer

    Dim ipStr As String = ""

    If
tagEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_TAG_REA
D)) Then
        Dim str As String
        Dim n As Integer
        n = CInt(Convert.ToInt16(tagEvent.tag.dataLen))
        For i As Integer = 0 To n - 1
            str = tagEvent.tag.data(i)
            ipStr += str + ""
        Next
    End If

    ReportTagEvent(tagEvent)

    Return 0
End Function

Private Sub ReportTagEvent(ByRef tagEvent As AW_API_NET.rfTagEvent_t)

    If tagEvent.tag.id <> 0 Then
        If tagEvent.tag.temp.temperature <> 0 Then
            'MessageBox.Show(tagEvent.tag.id.ToString + " " +
tagEvent.tag.temp.temperature.ToString)
            insertrow(tagEvent.tag.id.ToString(), tagEvent.tag.temp.temperature.ToString,
tagEvent.errorStatus.ToString())
            insertrow1(tagEvent.tag.id.ToString(), tagEvent.tag.temp.temperature.ToString,
tagEvent.errorStatus.ToString())
        End If

        If tagEvent.tag.temp.temperature > 38.9 Then
            MessageBox.Show(tagEvent.tag.id.ToString() + " " + "temperature very
high")
        End If
    End If
End Sub

```

```

        insertrow(tagEvent.tag.id.ToString(), tagEvent.tag.temp.temperature.ToString,
tagEvent.errorStatus.ToString())

```

```

    End If

```

```

End If

```

```

End Sub

```

```

Private Sub insertrow(ByVal tgid As Integer, ByVal temp As String, ByVal st As
Integer)

```

```

    Dim cn As New MySqlConnection(strCn)

```

```

    Dim cn1 As New MySqlConnection(strCn)

```

```

    Dim prvTime As DateTime = Nothing

```

```

    Dim prvTime1 As DateTime = Nothing

```

```

    Dim name = Nothing = ""

```

```

    Dim prvName As String = ""

```

```

    Dim prvIC_Num As String = ""

```

```

    Dim prvtagid As String = ""

```

```

    Dim prvDateofbirth As DateTime = Now

```

```

    Dim prvdateofadmin As DateTime = Now

```

```

    Dim prvDiagnosis As String = ""

```

```

    Dim prvTemperature As Integer = 0

```

```

    Dim prvage As Integer = 0

```

```

    If st = 0 Then

```

```

        Dim prvTimeCmd As New MySqlCommand("SELECT time FROM temptrack
WHERE tagID = " & tgid.ToString & "", cn)

```

```

        Dim prvTimeCmd3 As New MySqlCommand("SELECT name FROM
patint_details WHERE tagID = " & tgid.ToString & "", cn1)

```

```

        Dim prvTimeRdr As MySqlDataReader

```

```

        cn.Open()

```

```

        prvTimeRdr = prvTimeCmd.ExecuteReader

```

```

        While prvTimeRdr.Read

```

```

            prvTime = prvTimeRdr.Item("time")

```

```

        End While

```

```

        cn.Close()

```

```

        If (Format(prvTime, "yyyy-MM-dd HH:mm:ss") <
Format(DateTime.Now.AddSeconds(-20), "yyyy-MM-dd HH:mm:ss")) Then

```

```

            Dim insCmd As New MySqlCommand("INSERT INTO temptrack (tagID,
name, temperature, time) VALUES (" & tgid.ToString & ", " & name & ", " & temp &
", " & Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss") & ")", cn)

```

```

            Dim insCmd2 As New MySqlCommand("INSERT INTO archives (name,
ic_num, tagid, dateofbirth, dateofadmin, diagnosis, temperature, age, time) VALUES (" &
prvName & ", " & prvIC_Num & ", " & tgid.ToString & ", " & prvDateofbirth & ", " &

```

```
prvdateofadmin & "," & prvDiagnosis & "," & temp & "," & prvage & "," &
Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss") & ""), cn)
```

```
    cn.Open()
    insCmd.ExecuteNonQuery()
    insCmd2.ExecuteNonQuery()
    cn.Close()
    'MessageBox.Show(tgid.ToString + " " + temp)
End If
```

```
End If
```

```
End Sub
```

```
Private Sub insertrow1(ByVal tgid As Integer, ByVal temp As String, ByVal st As
Integer)
```

```
    'Dim cn As New MySqlConnection(strCn)
    Dim cn1 As New MySqlConnection(strCn)
```

```
    Dim prvTime As DateTime = Nothing
    'Dim prvTime1 As DateTime = Nothing
    Dim name = Nothing = ""
    Dim prvName As String = ""
    Dim prvIC_Num As String = ""
    Dim prvtagid As String = ""
    Dim prvDateofbirth As DateTime = Now
    Dim prvdateofadmin As DateTime = Now
    Dim prvDiagnosis As String = ""
    Dim prvTemperature As Integer = 0
    Dim prvage As Integer = 0
```

```
If st = 0 Then
```

```
    'Dim prvTimeCmd As New MySqlCommand("SELECT time FROM temptrack
WHERE tagID = " & tgid.ToString & "", cn)
    Dim prvTimeCmd As New MySqlCommand("SELECT * FROM patint_details
WHERE tagid_patid = " & tgid.ToString & "", cn1)
    'Dim prvTimeCmd2 As New MySqlCommand("SELECT time FROM temptrack
WHERE tagID = " & tgid.ToString & "", cn)
```

```
    Dim prvTimeRdr As MySqlDataReader
```

```
    '    cn.Open()
    '    prvTimeRdr = prvTimeCmd.ExecuteReader
    '
    'prvTimeRdr1 = prvTimeCmd1.ExecuteReader
    '    While prvTimeRdr.Read
    '    prvTime = prvTimeRdr.Item("time")
    ' End While
```

```
    'cn.Close()
```

```
    'Dim prvTimeRdr As MySqlDataReader
    cn1.Open()
```



```

prvTimeRdr = prvTimeCmd.ExecuteReader
prvTimeRdr = prvTimeCmd.ExecuteReader
' While prvTimeRdr.Read

' End While
While prvTimeRdr.Read
    prvTime = Now
    prvTimeRdr.Item("time")
    prvName = prvTimeRdr.Item("name")
    prvIC_Num = prvTimeRdr.Item("ic_num")
    prvtagid = prvTimeRdr.Item("tagid_patid")
    prvDateofbirth = prvTimeRdr.Item("dateofbirth")
    prvdateofadmin = prvTimeRdr.Item("dateofadmin")
    prvDiagnosis = prvTimeRdr.Item("diagnosis")
    prvTemperature = prvTimeRdr.Item("temperature")
    prvage = prvTimeRdr.Item("age")

End While
cn1.Close()

If (Format(prvTime, "yyyy-MM-dd HH:mm:ss") <
Format(DateTime.Now.AddSeconds(-20), "yyyy-MM-dd HH:mm:ss")) Then
    'Dim insCmd As New MySqlCommand("INSERT INTO temptrack (tagID, name,
temperature, time) VALUES (" & tgid.ToString & ", " & name & ", " & temp & ", " &
Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss") & ")", cn)

    ' Dim insCmd2 As New MySqlCommand("INSERT INTO archives (name,
ic_num, tagid, dateofbirth, dateofadmin, diagnosis, temperature, age, time) VALUES (" &
prvName & ", " & prvIC_Num & ", " & tgid.ToString & ", " & prvDateofbirth & ", " &
prvdateofadmin & ", " & prvDiagnosis & ", " & temp & ", " & prvage & ", " &
Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss") & ")", cn1)

    'cn.Open()
    'insCmd.ExecuteNonQuery()
    'insCmd2.ExecuteNonQuery()
    'cn.Close()
    'MessageBox.Show(tgid.ToString + " " + temp)
    'End If

If (Format(prvTime, "yyyy-MM-dd HH:mm:ss") <
Format(DateTime.Now.AddSeconds(-20), "yyyy-MM-dd HH:mm:ss")) Then
    'Dim insCmd As New MySqlCommand("INSERT INTO temptrack (tagID,
name, temperature, time) VALUES (" & tgid.ToString & ", " & name & ", " & temp &
", " & Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss") & ")", cn)

    Dim insCmd2 As New MySqlCommand("INSERT INTO archives (name,
ic_num, tagid, dateofbirth, dateofadmin, diagnosis, temperature, age, time) VALUES (" &
prvName & ", " & prvIC_Num & ", " & tgid.ToString & ", " & prvDateofbirth & ", " &
prvdateofadmin & ", " & prvDiagnosis & ", " & temp & ", " & prvage & ", " &
Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss") & ")", cn1)

    cn1.Open()

```

```

        'insCmd.ExecuteNonQuery()
        insCmd2.ExecuteNonQuery()
        cn1.Close()
        'MessageBox.Show(tgid.ToString + " " + temp)
    End If

End If
End Sub

Public Function GetStringIP(ByVal ip As Byte()) As String

    Dim p As Integer
    Dim s As String
    Dim ct As Integer

    ct = 0
    p = 0
    s = ""
    While (Convert.ToBoolean((ct <= 3)) AndAlso Convert.ToBoolean((p < 20))
AndAlso Convert.ToBoolean((ip(p) > 0)))
        If ip(p) > 46 Then
            s += Convert.ToString(ip(p) - 48) & "-"
            p += 1
        Else
            ct += 1
            p += 1
            s += "."
        End If
    End While
    Return s
End Function

Private Sub readConfig()
    Try
        ' Create an instance of StreamReader to read from a file.
        Using sr As StreamReader = New StreamReader("config.txt")
            Dim line As String
            ' Read and display the lines from the file until the end
            ' of the file is reached.
            frmConfig.lblIp.Items.Clear()
            line = sr.ReadLine()
            If Mid(line, 1, 3) = "ip:" Then
                frmConfig.lblIp.Items.Add(Mid(line, 4))
            End If
            'lblIp.Items.Clear()
            While Not line Is Nothing
                line = sr.ReadLine()
                If Mid(line, 1, 3) = "ip:" Then
                    frmConfig.lblIp.Items.Add(Mid(line, 4))
                End If
            End While
        End While
    End Try
End Sub

```

```

        sr.Close()
    End Using
Catch E As Exception
    ' Let the user know what went wrong.
    Console.WriteLine("The file could not be read:")
    Console.WriteLine(E.Message)
End Try

End Sub

Private Sub BtnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnConnect.Click
    scanIP()
    commStatus.BackColor = Color.DarkSeaGreen
End Sub

Private Sub scanIP()
    Dim iRet As Integer
    Dim ip(20) As Byte
    Dim stripx As String

    readIp()

    If (myPKTID >= 223) Then
        myPKTID = 1
    Else
        myPKTID = myPKTID + 1
    End If

    If registered = False Then

        ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)

        ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)

        registered = True
    End If

    For c As Integer = 0 To frmConfig.lbIp.Items.Count - 1
        stripx = frmConfig.lbIp.Items.Item(c).ToString

        For i As Integer = 0 To stripx.ToString.Length - 1
            ip(i) = Convert.ToByte(stripx.ToString.Chars(i))
        Next i
        iRet = ActiveWaveAPI.rfScanIP(ip, Convert.ToUInt16(myPKTID))

    Next c
    OpenConn()
End Sub

Private Sub OpenConn()
    Dim iRet As Integer

```

```

Dim ip(20) As Byte
Dim cIP(20) As Char
Dim stripx As String = "192.168.10.26"

If (myPKTID >= 223) Then
    myPKTID = 1
Else
    myPKTID = myPKTID + 1
End If
iRet = ActiveWaveAPI.rfOpenSocket(readerIP, readerPort, False,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS), Convert.ToUInt16(myPKTID))

If (myPKTID >= 223) Then
    myPKTID = 1
Else
    myPKTID = myPKTID + 1
End If
iRet = ActiveWaveAPI.rfResetReader(UInt16.Parse(1), UInt16.Parse(0),
UInt16.Parse(0), Convert.ToUInt16(AW_API_NET.APIConsts.ALL_READERS),
Convert.ToUInt16(myPKTID))

End Sub

Private Sub readIp()
Try
    Using sr As StreamReader = New StreamReader("config.txt")
        Dim line As String
        line = sr.ReadLine()
        If Mid(line, 1, 3) = "ip:" Then
            frmConfig.lblp.Items.Add(Mid(line, 4))
        End If
        While Not line Is Nothing
            line = sr.ReadLine()
            If Mid(line, 1, 3) = "ip:" Then
                frmConfig.lblp.Items.Add(Mid(line, 4))
            End If
        End While
        sr.Close()
    End Using
Catch E As Exception
    ' Let the user know what went wrong.
    Console.WriteLine("The file could not be read:")
    Console.WriteLine(E.Message)
End Try
End Sub

Private Sub CloseConn()

Dim iRet As Integer
Dim ip(20) As Byte
Dim cIP(20) As Char

```

```

If (myPKTID >= 223) Then
    myPKTID = 1
Else
    myPKTID = myPKTID + 1
End If

'If AllIPRadioButton.Checked Then
iRet = ActiveWaveAPI.rfCloseSocket(readerIP,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS))
IPListBox.Items.Clear()

End Sub

Private Sub btnConfig_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConfig.Click

    frmConfig.Show()

End Sub

Private Sub tmrTemp_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrTemp.Tick
    rqstTemp()
End Sub

Private Sub rqstTemp()
    Dim iRet As Integer
    Dim tagSelect As AW_API_NET.rfTagSelect_t
    Dim tagList(50) As UInt32
    Dim rdrID As UInt16
    Dim longInterval As Boolean
    Dim RdrCmdType As Integer

    rdrID = UInt16.Parse(0)

    RdrCmdType = APIConsts.ALL_READERS

    tagSelect.tagList = tagList

    tagSelect.tagList(0) = UInt32.Parse(0)
    tagSelect.numTags = Convert.ToUInt32(50)

    tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_FIELD)

    tagSelect.tagType = APIConsts.ACCESS_TAG

    If (myPKTID >= 223) Then
        myPKTID = 1
    Else
        myPKTID = myPKTID + 1
    End If

```

```

longInterval = False

'iRet = ActiveWaveAPI.rfCallTags(UInt16.Parse(1), rdrID, UInt16.Parse(0),
UInt16.Parse(0), tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))
'iRet = ActiveWaveAPI.rfQueryTags(UInt16.Parse(1), rdrID, UInt16.Parse(0),
tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))
iRet = ActiveWaveAPI.rfGetTagTemp(UInt16.Parse(1), rdrID, UInt16.Parse(0),
tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))

End Sub

Private Sub BtnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnStart.Click
    tmrTemp.Enabled = True
End Sub

Private Sub BtnHide_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnHide.Click
    Me.Hide()
End Sub
End Class

```

iv. **System administration**

```
Public Class System_admin
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
```

```
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Close()
    End If
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Back to Main Menu ?", "Back Conformation", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Hide()
        Main_Menu.Show()
    End If

```

End Sub

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Add New User ?", "Add User", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Hide()
        add_admin.Show()
    End If
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Add New Patient", "Add Patient", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Hide()
        Patient_reg.Show()
    End If
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    Me.Hide()
    frmDelete.Show()
End Sub
End Class
```

v. **Admin Registration**

```
Imports MySql.Data.MySqlClient
Public Class add_admin
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
```

```
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Close()
    End If
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
```

```
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Back to Main Menu ?", "Back Conformation", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Hide()
        Main_Menu.Show()
    End If
```

```
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
```

```
    txtIDnum.Clear()
    txtName.Clear()
    'txtPswd1.Clear()
    txtpswd.Clear()
    txtUname.Clear()
    txtIDnum.Focus()
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
    Dim name As String = txtName.Text.ToString
    Dim rowsCheck As Integer = 0
    Dim ID As String = txtIDnum.Text.ToString
    Dim Post As String = cboPost.Text.ToString
    Dim user_name As String = txtUname.Text.ToString
    Dim Password As String = txtpswd.Text.ToString
    Dim dochp As String = txtHP.Text
    Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
```

```
    'Connect to database
```

```
    Dim cn As New MySqlConnection(strCn)
```

```
    'Open database
```

```
    Dim cmdCheck As New MySqlCommand("SELECT * FROM registration WHERE
user_name = " & user_name & "'", cn)
```

```
    Dim daCheck As New MySqlDataAdapter
```

```
    Dim dsCheck As New DataSet
```

```
    Dim dtCheck As New DataTable
```

```
    cn.Open()
```

```
    'Check from table (search)
```

```
    Try
```

```
        With daCheck
```

```
            .SelectCommand = cmdCheck
```



```

        .Fill(dsCheck, "searchresult")
    End With
    dtCheck = dsCheck.Tables("searchresult")
    rowsCheck = dtCheck.Rows.Count()

    'If unable to connect, show error!
    Catch ex As Exception
        MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!!")
    End Try

    'If already connected, close connection
    If CBool(ConnectionString.Open) Then
        cn.Close()
    End If

    If rowsCheck = 0 Then
        'Open connection
        cn.Open()

        'Check for name and if no existing inmates available, add them
        Dim cmdIns As New MySqlCommand("INSERT INTO registration (ID, Post,
user_name, Password, name, DoctorHP) VALUES ('" & ID & "','" & Post & "','" &
user_name & "','" & Password & "','" & name & "','" & dochp & "')", cn)
        cmdIns.ExecuteNonQuery()

        If CBool(ConnectionString.Open) Then
            cn.Close()
        End If

        'If registration successful, show ID and name
        MsgBox(name & "/" & ID & " has been registered successfully")

    Else
        'If already existing officer, show error
        MsgBox("Error!!! " & name & "is already registered in the database")

    End If

End Sub
End Class

```

vi. Patient registration

```

Imports System.Data
Imports MySql.Data.MySqlClient

Public Class Patient_reg

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

```

```

Dim drExit As DialogResult
drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
    MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
    MessageBoxDefaultButton.Button2)
If drExit = DialogResult.Yes Then
    Me.Close()
End If
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Back to Main Menu ?", "Back Conformation", _
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
        Me.Hide()
        Main_Menu.Show()
    End If
End Sub

```

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    txtAge.Clear()
    txtdateofadmit.Clear()
    txtDiag.Clear()
    txtDob.Clear()
    txtICnum.Clear()
    txtName.Clear()
    txtPatID.Clear()
    txtTemp.Clear()
    cboTagID.ResetText()
    txtAge.Focus()

End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim name As String = txtName.Text.ToString
    Dim rowsCheck As Integer = 0
    Dim icnum As String = txtICnum.Text.ToString
    Dim diag As String = txtDiag.Text.ToString
    Dim dob As String = txtDob.Text.ToString
    Dim temp As String = txtTemp.Text.ToString
    Dim tagid As String = cboTagID.Text.ToString
    Dim age As String = txtPatID.Text.ToString
    Dim doa As DateTime = DateTime.Now
    Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="

    'Connect to database
    Dim cn As New MySqlConnection(strCn)

```

```

'Open database
Dim cmdCheck As New MySqlCommand("SELECT * FROM patint_details
WHERE tagid_patid = '" & tagid & "'", cn)
Dim daCheck As New MySqlDataAdapter
Dim dsCheck As New DataSet
Dim dtCheck As New DataTable

cn.Open()

'Check from table (search)
Try
    With daCheck
        .SelectCommand = cmdCheck
        .Fill(dsCheck, "searchresult")
    End With
    dtCheck = dsCheck.Tables("searchresult")
    rowsCheck = dtCheck.Rows.Count()

    'If unable to connect, show error!
Catch ex As Exception
    MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
End Try

'If already connected, close connection
If CBool(ConnectionString.Open) Then
    cn.Close()
End If

If rowsCheck = 0 Then
    'Open connection
    cn.Open()

    'Check for name and if no existing inmates available, add them
    Dim cmdIns As New MySqlCommand("INSERT INTO patint_details (name,
ic_num, tagid_patid, dateofbirth, dateofadmin, diagnosis, temperature, age) VALUES ('"
& name & "','" & icnum & "','" & tagid & "','" & dob & "','" & Format(DateTime.Now,
"yyyy-MM-dd HH:mm:ss") & "','" & diag & "','" & temp & "','" & age & "')", cn)
    Dim cmdIns1 As New MySqlCommand("INSERT INTO archive_names (name,
ic_num, tagid, dateofbirth, dateofadmin, diagnosis, temperature, age) VALUES ('"
& name & "','" & icnum & "','" & tagid & "','" & dob & "','" & Format(DateTime.Now,
"yyyy-MM-dd HH:mm:ss") & "','" & diag & "','" & temp & "','" & age & "')", cn)

    cmdIns.ExecuteNonQuery()
    cmdIns1.ExecuteNonQuery()
    If CBool(ConnectionString.Open) Then
        cn.Close()
    End If

    'If registration successful, show ID and name
    MsgBox(name + "/" + icnum + " has been registered successfully")

```

```

Else
    'If already existing officer, show error
    MsgBox("Error!!! " + name + "is already registered in the database")

End If
End Sub

Private Sub Patient_reg_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub

Private Sub cboTagID_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cboTagID.SelectedIndexChanged

End Sub
End Class

```

vii. Record Menu

```

Public Class records_menu

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Me.Hide()
    Patient_Records.Show()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Me.Hide()
    achives_data.Show()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Me.Hide()
    Main_Menu.Show()
End Sub
End Class

```

viii. Patient Record

```

Imports System.Data
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient

```

Imports System.Threading

Public Class Patient_Records

Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="

Dim temper As ListViewItem

Dim tgID As String

Private str2 As System.Threading.Thread

Private str1 As System.Threading.Thread

Private Sub Patient_Records_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = False

lvwTemp.Columns.Clear()

lvwTemp.Columns.Add("Temperature", 100, HorizontalAlignment.Left)

lvwTemp.Columns.Add("Date
Time", 225,
HorizontalAlignment.Right)

lvwTemp.Hide()

End Sub

Private Sub listDetails()

Dim name As String = ""

Dim diagn As String = ""

Dim birthD As DateTime = Nothing

Dim age As Integer = 0

Dim admisD As DateTime = Nothing

Dim ic As String = ""

tgID = txtID.Text

If (checkID(tgID) = True) Then

If Me.InvokeRequired Then

Me.Invoke(New MethodInvoker(AddressOf listDetails))

Else

Establish connection

Dim temp As String = ""

Dim dat As DateTime = Nothing

Dim cn As New MySqlConnection(strCn)

'Fill in Patient details

Dim cmdDetails As New MySqlCommand("SELECT * FROM patint_details
WHERE tagid_patid = " & tgID & "", cn)

Dim rdrDetails As MySqlDataReader

```

cn.Open()
rdrDetails = cmdDetails.ExecuteReader
While rdrDetails.Read
    name = rdrDetails.Item("name")
    ic = rdrDetails.Item("ic_num")
    birthD = rdrDetails.Item("dateofbirth")
    diagn = rdrDetails.Item("diagnosis")
    age = rdrDetails.Item("age")
    admisD = rdrDetails.Item("dateofadmin")
End While

rdrDetails.Close()
cn.Close()

txtname.Text = name
txtage.Text = age.ToString
txtdiag.Text = diagn
txtdoa.Text = Format(admisD, "yyyy-MM-dd HH:mm:ss")
txtdob.Text = Format(birthD, "yyyy-MM-dd HH:mm:ss")
txticnum.Text = ic

txtname.Show()
txtage.Show()
txtdoa.Show()
txtdiag.Show()
txticnum.Show()
txtdob.Show()
Label3.Show()
Label4.Show()
Label5.Show()
Label6.Show()
Label7.Show()
Label8.Show()

btnConfirm.Show()

End If

Else

MessageBox.Show("There is no record corresponding to this ID")
txtname.Text = ""
txtage.Text = ""
txtdoa.Text = ""
txtdiag.Text = ""
txticnum.Text = ""
txtdob.Text = ""
lvwTemp.Items.Clear()
lvwTemp.Hide()
btnConfirm.Hide()
End If

```

```

End Sub

Sub Thread1()
    listDetails()
End Sub

Private Function checkID(ByVal id As String) As Boolean

    Dim found As Boolean = False
    Dim rows As Integer = 0
    Dim myConnString As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
    Dim cn As New MySqlConnection(myConnString)
    Dim cmdGH As New MySqlCommand("SELECT * FROM patint_details WHERE
tagid_patid = '" & id & "'", cn)
    Dim daGH As New MySqlDataAdapter
    Dim dsGH As New DataSet
    Dim dtGH As New DataTable

    cn.Open()

    Try
        daGH.SelectCommand = cmdGH
        daGH.Fill(dsGH, "details")
        dtGH = dsGH.Tables("details")
        rows = dtGH.Rows.Count()
    Catch ex As Exception
        MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
    End Try

    If ConnectionState.Open Then
        cn.Close()
    End If

    If (rows > 0) Then
        found = True
    Else
        found = False
    End If

    Return found

End Function

Private Sub listvw()

    Dim i As Integer = 0

    tglID = txtID.Text
    Dim cn As New MySqlConnection(strCn)
    If Me.InvokeRequired Then

```

```

        Me.Invoke(New MethodInvoker(AddressOf listvw))
    Else

        'Populate the ListView
        Dim cmd As New MySqlCommand("SELECT * FROM temprack WHERE tagID
= " & tgID & "", cn)
        Dim rdrTemp As MySqlDataReader

        lvwTemp.Items.Clear()

        If cn.State = ConnectionState.Closed Then
            cn.Open()
        End If

        'Display raw in listview (attendlist)
        rdrTemp = cmd.ExecuteReader()
        While rdrTemp.Read
            temper = lvwTemp.Items.Add(rdrTemp.Item("temperature"))
            temper.SubItems.Add(Format(rdrTemp.Item("time"), "yyyy-MM-dd
HH:mm:ss"))
        End While

        rdrTemp.Close()
        cn.Close()

        'coloring background
        While i <= lvwTemp.Items.Count - 1
            If i Mod 2 = 0 Then
                lvwTemp.Items(i).BackColor = Color.Aquamarine
            Else
                lvwTemp.Items(i).BackColor = Color.White
            End If
            i = i + 1
        End While

        lvwTemp.Show()

    End If

End Sub

Sub Thread2()
    listvw()
End Sub

Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSearch.Click
    str1 = New Thread(AddressOf Thread1)
    str1.Start()
End Sub

```



```

Private Sub btnConfirm_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConfirm.Click
    str2 = New Thread(AddressOf Thread2)
    str2.Start()
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Main_Menu.Show()
    Me.Hide()
End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Me.Close()
End Sub
End Class

```

ix. Archive Data

```

Imports System.Data
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient
Imports System.Threading

```

```

Public Class achives_data

```

```

    Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
    Dim temper As ListViewItem
    Dim tgID As String
    Private str2 As System.Threading.Thread
    Private str1 As System.Threading.Thread

```

```

Private Sub achives_data_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

```

```

    System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = False

```

```

    lvwTemp.Columns.Clear()
    lvwTemp.Columns.Add("Temperature", 100, HorizontalAlignment.Left)
    lvwTemp.Columns.Add("Date Time", 225,
HorizontalAlignment.Right)

```

```

    lvwTemp.Hide()

```

```

End Sub

```

Private Sub listDetails()

```
Dim name As String = ""
Dim diagn As String = ""
Dim birthD As DateTime = Nothing
Dim age As Integer = 0
Dim admisD As DateTime = Nothing
Dim ic As String = ""
```

tgID = txtID.Text

If (checkID(tgID) = True) Then

```
If Me.InvokeRequired Then
    Me.Invoke(New MethodInvoker(AddressOf listDetails))
Else
```

```
'Establish connection
Dim temp As String = ""
Dim dat As DateTime = Nothing
Dim cn As New MySqlConnection(strCn)
```

```
'Fill in Patient details
Dim cmdDetails As New MySqlCommand("SELECT * FROM archive_names
WHERE tagid = " & tgID & "", cn)
Dim rdrDetails As MySqlDataReader
```

```
cn.Open()
rdrDetails = cmdDetails.ExecuteReader
While rdrDetails.Read
    name = rdrDetails.Item("name")
    ic = rdrDetails.Item("ic_num")
    birthD = rdrDetails.Item("dateofbirth")
    diagn = rdrDetails.Item("diagnosis")
    age = rdrDetails.Item("age")
    admisD = rdrDetails.Item("dateofadmin")
End While
```

```
rdrDetails.Close()
cn.Close()
```

```
txtname.Text = name
txtage.Text = age.ToString
txtdiag.Text = diagn
txtdoa.Text = Format(admisD, "yyyy-MM-dd HH:mm:ss")
txtdob.Text = Format(birthD, "yyyy-MM-dd HH:mm:ss")
txticnum.Text = ic
```

```
txtname.Show()
txtage.Show()
txtdoa.Show()
txtdiag.Show()
```

```

        txticnum.Show()
        txtdob.Show()
        Label3.Show()
        Label4.Show()
        Label5.Show()
        Label6.Show()
        Label7.Show()
        Label8.Show()

        btnConfirm.Show()

    End If

Else

    MessageBox.Show("There is no record corresponding to this ID")
    txtname.Text = ""
    txtage.Text = ""
    txtdoa.Text = ""
    txtdiag.Text = ""
    txticnum.Text = ""
    txtdob.Text = ""
    lvwTemp.Items.Clear()
    lvwTemp.Hide()
    btnConfirm.Hide()
End If

End Sub

Sub Thread1()
    listDetails()
End Sub

Private Function checkID(ByVal id As String) As Boolean

    Dim found As Boolean = False
    Dim rows As Integer = 0
    Dim myConnString As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
    Dim cn As New MySqlConnection(myConnString)
    Dim cmdGH As New MySqlCommand("SELECT * FROM archive_names WHERE
tagid = " & id & """, cn)
    Dim daGH As New MySqlDataAdapter
    Dim dsGH As New DataSet
    Dim dtGH As New DataTable

    cn.Open()

    Try
        daGH.SelectCommand = cmdGH
        daGH.Fill(dsGH, "details")
        dtGH = dsGH.Tables("details")
    
```

```

        rows = dtGH.Rows.Count()
    Catch ex As Exception
        MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
    End Try

    If ConnectionState.Open Then
        cn.Close()
    End If

    If (rows > 0) Then
        found = True
    Else
        found = False
    End If

    Return found

End Function

Private Sub listvw()

    Dim i As Integer = 0

    tgID = txtID.Text
    Dim cn As New MySqlConnection(strCn)
    If Me.InvokeRequired Then
        Me.Invoke(New MethodInvoker(AddressOf listvw))
    Else

        'Populate the ListView
        Dim cmd As New MySqlCommand("SELECT * FROM archives WHERE tagID =
"" & tgID & """, cn)
        Dim rdrTemp As MySqlDataReader

        lvwTemp.Items.Clear()

        If cn.State = ConnectionState.Closed Then
            cn.Open()
        End If

        'Display raw in listview (attendlist)
        rdrTemp = cmd.ExecuteReader()
        While rdrTemp.Read
            temper = lvwTemp.Items.Add(rdrTemp.Item("temperature"))
            temper.SubItems.Add(Format(rdrTemp.Item("time"), "yyyy-MM-dd
HH:mm:ss"))
        End While

        rdrTemp.Close()
        cn.Close()
    End If
End Sub

```

```

'coloring background
While i <= lvwTemp.Items.Count - 1
    If i Mod 2 = 0 Then
        lvwTemp.Items(i).BackColor = Color.Aquamarine
    Else
        lvwTemp.Items(i).BackColor = Color.White
    End If
    i = i + 1
End While

lvwTemp.Show()

End If

End Sub

Sub Thread2()
    listvw()
End Sub

Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSearch.Click
    str1 = New Thread(AddressOf Thread1)
    str1.Start()
End Sub

Private Sub btnConfirm_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConfirm.Click
    str2 = New Thread(AddressOf Thread2)
    str2.Start()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Main_Menu.Show()
    Me.Hide()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Me.Close()
End Sub
End Class

```

x. **SMS Alert**

```
Imports System.Data
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient
Imports System.Threading
```

Public Class Details

```
    Dim strCn As String = "Database=hospital_db;Data
Source=localhost;User Id=root;Password="
    Dim ID As String
    Dim Dname As String
    Dim DrHP As String
    Private str2 As System.Threading.Thread
    Private str1 As System.Threading.Thread
    Dim cn As New MySqlConnection(strCn)
    Dim cn1 As New MySqlConnection(strCn)
    Dim cn2 As New MySqlConnection(strCn)
    Dim rdrDetails As MySqlDataReader
    Dim rdrDetails1 As MySqlDataReader
    Dim rdrDetails2 As MySqlDataReader
    Private Function checkID(ByVal id As String) As Boolean
```

```
        Dim found As Boolean = False
        Dim rows As Integer = 0
        Dim myConnString As String = "Database=hospital_db;Data
Source=localhost;User Id=root;Password="
        Dim cn3 As New MySqlConnection(myConnString)
        Dim cmdGH As New MySqlCommand("SELECT * FROM
patint_details WHERE tagid_patid = '" & id & "'", cn3)
        Dim daGH As New MySqlDataAdapter
        Dim dsGH As New DataSet
        Dim dtGH As New DataTable
```

```
        cn3.Open()
```

```
        Try
```

```
            daGH.SelectCommand = cmdGH
            daGH.Fill(dsGH, "details")
            dtGH = dsGH.Tables("details")
            rows = dtGH.Rows.Count()
```

```
        Catch ex As Exception
```

```
            MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try
```

```
        If ConnectionState.Open Then
```

```
            cn3.Close()
```

```
        End If
```

```
        If (rows > 0) Then
```

```
            found = True
```

```

Else
    found = False
End If

Return found

End Function

Private Sub btnCheck_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCheck.Click
    Dim cmdDetails As New MySqlCommand("SELECT * FROM
registration WHERE id = '" & txtID.Text & "'", cn)
    cn.Open()

    rdrDetails = cmdDetails.ExecuteReader
    While rdrDetails.Read
        Dname = rdrDetails.Item("name")
        DrHP = rdrDetails.Item("DoctorHP")
    End While
    rdrDetails.Close()
    cn.Close()

    Dim cmdSendData As New MySqlCommand("SELECT * FROM
patint_details WHERE tagid_patid = '" & txtpatID.Text & "'",
cn1)

    Dim Message As Object
    Dim pname As String = ""
    Dim icnum As String = ""
    Dim doa As String = ""
    Dim diagnosis As String = ""
    Dim age As Integer

    cn1.Open()

    rdrDetails1 = cmdSendData.ExecuteReader

    While rdrDetails1.Read
        pname = rdrDetails1.Item("name")
        icnum = rdrDetails1.Item("ic_num")
        doa = rdrDetails1.Item("dateofadmin")
        diagnosis = rdrDetails1.Item("diagnosis")
        age = rdrDetails1.Item("age")
    End While

    rdrDetails1.Close()
    cn1.Close()

    Message = " ATTENTION The Patient: " & pname & " Age "
& age & " Id Number " & icnum & " Admitted on " & doa & "
Diagonised with " & diagnosis & " Temperature is Critical "

    lblAlert.Text = Message
    Shapel.FillColor = System.Drawing.Color.Blue
    Shape2.FillColor = System.Drawing.Color.Blue
    txtName.Text = Dname
    txtHPNum.Text = DrHP

```

```

btnConfirm.Show()
btnBack.Show()
txtName.Show()
Label4.Show()
Label2.Show()
Label5.Show()
LblAlert.Show()
txtHPNum.Show()
txtpatID.Show()

If txtID.Text = "" Then
    LblAlert.Hide()
    btnConfirm.Hide()
    btnBack.Hide()
    txtName.Hide()
    Label4.Hide()
    Label2.Hide()
    txtHPNum.Hide()
    Shape1.FillColor = System.Drawing.Color.Red
    Shape2.FillColor = System.Drawing.Color.Red
    txtID.BackColor = System.Drawing.Color.Red
    MsgBox("Please enter Doctor ID to continue")
    txtID.Focus()
ElseIf txtpatID.Text = "" Then
    LblAlert.Hide()
    btnConfirm.Hide()
    btnBack.Hide()
    txtName.Hide()
    Label4.Hide()
    Label2.Hide()
    txtHPNum.Hide()
    Shape1.FillColor = System.Drawing.Color.Red
    Shape2.FillColor = System.Drawing.Color.Red
    txtpatID.BackColor = System.Drawing.Color.Red
    MsgBox("Please enter Patient ID to continue")
    txtpatID.Focus()
ElseIf Dname = "" Then
    LblAlert.Hide()
    btnConfirm.Hide()
    btnBack.Hide()
    txtName.Hide()
    Label4.Hide()
    Label2.Hide()
    txtHPNum.Hide()
    Shape1.FillColor = System.Drawing.Color.Red
    Shape2.FillColor = System.Drawing.Color.Red
    MsgBox("Enter Valid Doctor ID")
    txtID.BackColor = System.Drawing.Color.Red
    txtID.Text = ""
    txtID.Focus()
ElseIf DrHP = "" Then
    LblAlert.Hide()
    btnConfirm.Hide()
    btnBack.Hide()
    txtName.Hide()
    Label4.Hide()

```



```

        Label2.Hide()
        txtHPNum.Hide()
        Shape1.FillColor = System.Drawing.Color.Red
        Shape2.FillColor = System.Drawing.Color.Red
        MsgBox("Enter Valid Doctor ID Doctor Phone Number
Missing")
        txtID.BackColor = System.Drawing.Color.Red
        txtID.Text = ""
        txtID.Focus()
    ElseIf pname = "" Then
        lblAlert.Hide()
        btnConfirm.Hide()
        btnBack.Hide()
        txtName.Hide()
        Label4.Hide()
        Label2.Hide()
        txtHPNum.Hide()
        Shape1.FillColor = System.Drawing.Color.Red
        Shape2.FillColor = System.Drawing.Color.Red
        MsgBox("Enter Valid Patient ID to continue")
        txtpatID.BackColor = System.Drawing.Color.Red
        txtpatID.Text = ""
        txtpatID.Focus()

    End If

End Sub

Private Sub btnConfirm_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnConfirm.Click
    Dim cmdSendData As New MySqlCommand("SELECT * FROM
patint_details WHERE tagid_patid = '" & txtpatID.Text & "'",
cn1)

    Dim Message As Object
    Dim pname As String = ""
    Dim icnum As String = ""
    Dim doa As String = ""
    Dim diagnosis As String = ""
    Dim age As Integer
    Dim callno As String
    'Dim Details() As String
    'Details = cmdSendData()
    cn1.Open()
    'rdrDetails = cmdDetails.ExecuteReader
    rdrDetails1 = cmdSendData.ExecuteReader

    While rdrDetails1.Read
        pname = rdrDetails1.Item("name")
        icnum = rdrDetails1.Item("ic_num")
        doa = rdrDetails1.Item("dateofadmin")
        diagnosis = rdrDetails1.Item("diagnosis")
        age = rdrDetails1.Item("age")
    End While

    rdrDetails1.Close()
    cn1.Close()

```

```
Message = " ATTENTION The Patient: " & pname & " Age "  
& age & " Id Number " & icnum & " Admitted on " & doa & "  
Diagonised with " & diagnosis & " Temperature is Critical "
```

```
LblAlert.Text = Message  
LblAlert.Visible = True  
callno = txtHPNum.Text
```

```
MSComm1.CommPort = 16 'switch on the port, change  
accordingly with your PC  
MSComm1.PortOpen = True
```

```
Shape1.FillColor = System.Drawing.Color.Blue  
Shape2.FillColor = System.Drawing.Color.Blue  
MSComm1.Settings = "9600,n,8,1"  
MSComm1.Output = "AT+CMGF=1" & vbCrLf  
MSComm1.Output = "AT+CMGS=" & Chr(34) & callno &  
Chr(34) & vbCrLf  
MSComm1.Output = Message & Chr(26)  
MSComm1.PortOpen = False
```

```
MsgBox("Message Sent")  
'ConnStart.Enabled = True  
'Me.Close()  
LblAlert.Hide()  
btnConfirm.Hide()  
btnBack.Hide()  
txtName.Hide()  
Label4.Hide()  
Label2.Hide()  
txtHPNum.Hide()  
Shape1.FillColor = System.Drawing.Color.Red  
Shape2.FillColor = System.Drawing.Color.Red
```

```
End Sub
```

```
End Class
```